ELSEVIER
Theoretical Computer Science 280 (2002) 163–192
Theoretical Computer Science
www.elsevier.com/locate/tcs

Compositional SOS and beyond: a coalgebraic view of open systems ☆

Andrea Corradini[a], Reiko Heckel[b,*], Ugo Montanari[a]

[a] *Dipartimento di Informatica, Università degli Studi di Pisa, Corso Italia, 40, I-56125 Pisa, Italy*
[b] *Universität Paderborn, FB 17 Mathematik und Informatik, Warburgerstrasse 100, D-33098 Paderborn, Germany*

Abstract

In this paper we address the issue of providing a structured coalgebra presentation of transition systems with algebraic structure on states determined by an equational specification Γ. More precisely, we aim at representing such systems as coalgebras for an endofunctor on the category of Γ-algebras. The systems we consider are specified by using arbitrary SOS rules, which in general do not guarantee that bisimilarity is a congruence. We first show that the structured coalgebra representation works only for systems where transitions out of complex states can be derived from transitions out of corresponding component states. This decomposition property of transitions indeed ensures that bisimilarity is a congruence. For a system not satisfying this requirement, next we propose a closure construction which adds *context transitions*, i.e., transitions that spontaneously embed a state into a bigger context or vice versa. The notion of bisimulation for the enriched system coincides with the notion of *dynamic bisimilarity* for the original one, i.e., with the coarsest bisimulation which is a congruence. This is sufficient to ensure that the structured coalgebra representation works for the systems obtained as result of the closure construction.

1. Introduction

Structured operational semantics (SOS) [27] is a simple and powerful style of language specification, where each language construct is defined separately by a few clauses. Most of the developments in the area of process algebras are based on SOS specifications, but often also functional and higher-order calculi and languages take advantage of them. Special formats have been defined (see e.g. [2, 9, 16]), which automatically guarantee important properties, like that bisimulation is a congruence for

☆ Research partially supported by the TMR Network *General Theory of Graph Transformation Systems* (*GETGRATS*), by the Esprit WG *Application of Graph Transformation* (*APPLIGRAPH*), and by the MURST project *Teoria della Concorrenza*, *Linguaggi di Ordine Superiore e Strutture di Tipi* (*TOSCA*).
* Corresponding author.
E-mail addresses: andrea@di.unipi.it (A. Corradini), reiko@uni-paderborn.de (R. Heckel), ugo@di.unipi.it (U. Montanari).

PII: S0304-3975(01)00025-1

the calculus under definition, or that a reduction system can be automatically derived from the SOS rules.

Possible limitations of the ordinary SOS approach are that little model theory has been actually developed, and that format restrictions exclude some of the most interesting calculi, like the π-calculus [1] [24]. Both limitations stem from the proof-theoretic point of view of the SOS approach to operational semantics, which exploits structural axioms only to a limited extent and is mainly interested in the initial model.

The usual model associated to an SOS specification is a labelled transition system, which can be easily seen as a coalgebra for an endofunctor in the category **Set**. However, in this representation the states of the system are seen just as forming a set, i.e., the algebraic structure modelling the construction of programs and the composition of states is disregarded. The relevance of the algebraic structure on states for the theory of transitions systems is witnessed, for example, by the problem of "bisimilarity as a congruence", which is essential for making compositional the abstract semantics based on bisimilarity.

The missing structure may be recovered by integrating coalgebras with algebras as it is done, for example, in [29]. In this categorical approach, the algebraic structure is represented by a monad T and the coalgebraic structure is given by a comonad D on the same base category. Then, *bialgebras* are defined as algebra–coalgebra pairs over a common carrier subject to a pentagonal law which ensures that the same structure can be seen both as a coalgebra in the category of T-algebras and as an algebra in the category of D-coalgebras. The corresponding liftings of the comonad D and the monad T to the T-algebras and D-coalgebras, respectively, are derived from a *distributive law* $\lambda : TD \Rightarrow DT$ which expresses the relation between the two structures. It is shown in [29] that such distributive law may be derived from a specification in GSOS format [2], a format which makes sure that bisimilarity is a congruence. In fact, the same is true for the overall framework of bialgebras since morphisms between bialgebras are both algebra homomorphisms and coalgebra morphisms, and thus the unique morphism to the final bialgebra, which exists under reasonable assumptions, induces a (coarsest) bisimulation congruence on any coalgebra.

A more concrete and (we believe) simpler presentation of essentially the same structure is introduced in [5]. Besides restricting on the coalgebraic side to coalgebras for an endofunctor, the algebraic structure is represented by an equational algebraic specification which, in contrast to the abstract categorical notion of monad, provides us with a concrete specification language. In fact, although the semantical framework of bialgebras allows to deal with algebras for an equational specification $\Gamma = \langle \Sigma, E \rangle$, the approach in [29] (like the GSOS format) is restricted to algebras for a signature Σ.

In our view, the development of [5] fits quite naturally into an approach that we can call of *structured models*, which is based on internal constructions. The idea is that *basic* models are built using sets and functions, and morphisms between basic

[1] A version of the π-calculus (without the replication operator) which fits in deSimone format, and thus for which a head-normalising axiom system can be immediately derived, is described in [11].

models are defined in terms of functions and of axioms represented as diagrams in **Set**. By replacing **Set** with an *environment* category **C** we can have automatically models enriched with the structure specified by **C**.

The structured model approach has been quite successful for *structured transition systems* [8], where the basic versions are defined as sets of states, sets of transitions and pairs of functions (i.e., source and target) between them. [2] In fact, just by varying the environment category, structured transition systems exactly describe such diverse models of computation as P/T Petri nets in the sense of [21], concurrent grammars, concurrent term rewriting, term graph rewriting [4], graph rewriting [6, 17], and Horn Clause Logic [3]. More interestingly, the free functor (which exists under mild conditions on **C**) mapping the category of structured transition systems on **C** to the category of internal categories in **C** actually corresponds to defining the operational semantics of these models of computation. Another related example is described in [25], where the notion of bisimilarity of [18] based on spans of open maps, initially defined for ordinary transition systems, is automatically lifted to certain *history dependent* transition systems which model name generation and name passing as necessary for the π-calculus.

In general, internal constructions can be defined using sketches [19] or using extensions of algebraic theories which allow for partial algebras like categories (see e.g. [20]), where internal constructions are represented as *tensor* composition of theories [15]. For instance, the theory of double categories, which are internal categories in **Cat**, can be defined as the tensor product of the theory of categories with itself [22].

Following the structured model approach, in this paper we want to study under which conditions transition systems can be represented as structured coalgebras on an environment category of algebras. We formalise general (positive) SOS rules as finite implications (Horn clauses) specifying a family of transition relations $\xrightarrow{l}_{l\in L}$ where L is a set of labels. This automatically defines a notion of generated transition system as the initial object in the category of systems satisfying the rules.

We consider transition systems where the collection of states is an algebra with respect to an equational specification $\Gamma = \langle \Sigma, E\rangle$, and where transitions are specified using general SOS rules. This allows us to consider also several of the rules which have been actually proposed in the literature and which cannot be handled by "well-behaved" formats. These include for instance the rules of the π-calculus by axiomatising substitution, and also axioms like

$$a.p \mid \bar{a}.q \xrightarrow{\tau} p \mid q$$

which is typical for the CHAM approach to operational semantics [1], but does not fit in any of the ordinary SOS formats since it applies to a complex term.

Unlike the GSOS format considered in [29], general SOS rules over an algebraic specification do not ensure that in the generated transition system bisimilarity is a congruence with respect to the operators defined on states, which is a necessary

[2] Labels on transitions and initial and final states can also be easily added.

condition for representing a system as a structured coalgebra. Therefore the abstract, categorical framework introduced in [29] is not directly applicable to our systems. Instead, following a similar intuition, we propose for a given set of SOS rules a construction which maps an algebra of states to a corresponding algebra over the sets of possible transitions from those states. We show that this construction defines a functor if one considers among the rules derivable from the given SOS specification only those which are in a specific well-behaved format, namely pure, look-ahead-free TYFT [16]. Furthermore, the equations defining the algebra of states must satisfy a suitable condition, that we call TYFT *bisimilarity*.

Based on this construction, our result is that, for representing a transition system $\langle S, \rightarrow \rangle$ satisfying the rules as a coalgebra in the category of Γ-algebras (where we assume that the equations satisfy the relevant conditions), the following condition is necessary and sufficient. There exists a transition $f^A(a_1, \ldots, a_n) \xrightarrow{l} b$ out of a composed state if and only if there is a TYFT *proof* for the transition (i.e., a proof using only pure, look-ahead-free TYFT rules derivable from the SOS specification) using as premises transitions out of the component states $a_1, \ldots, a_n$. That means, a specification with general SOS rules which is not equivalent to a specification with rules in this restricted TYFT format excludes the structured coalgebra interpretation of the generated transition system. Thus one could say that what was considered a methodological convenience, i.e., that in the SOS approach each language construct is defined separately by a few clauses, is in fact mandatory to guarantee a satisfactory algebraic structure.

The second part of the paper considers a rather different class of systems, but eventually, as a kind of side effect, solves the lifting problem for a class of transition systems which do not satisfy the condition above. *Open* systems are nowadays very important in distributed and network computing. One of their fundamental properties is the ability of adapting to additions of new components without requiring repeated compilations and initialisations. Thus for two open systems to be equivalent, not only experiments based on communications with the external world should be considered, but also experiments consisting of the additions of new components. In our setting, this corresponds to allow an extra clause in the definition of bisimulation where arbitrary contexts are applied. The resulting notion of equivalence has been considered in [26] and called *dynamic* bisimilarity. Of course, when ordinary bisimilarity is a congruence, dynamic bisimilarity coincides with it. In any case it can be characterised as the coarsest bisimulation which is a congruence. Dynamic bisimilarity is a rather stable notion, and can be defined in several equivalent ways. For CCS with unobservable τ transitions it does not coincide with observational congruence (which is not a bisimulation), but it is finer, and it can be axiomatised just by deleting one of Milner's τ laws.

Our result about open systems is that they fit our structured coalgebra characterisation if they can be provided with a set of universal contexts satisfying suitable conditions. More precisely, given an SOS specification with such a set of universal contexts, we can define its *context closure*, i.e., another specification including also the possible *context transitions*, namely all transitions resulting in the addition of some context and labelled by it. We prove that dynamic bisimilarity for the given specification coincides with ordinary bisimilarity for its context closure. In addition, any context closure can be seen as a structured coalgebra. Thus open systems, for which dynamic bisimilarity is the natural notion, always have a satisfactory algebraic structure. Ordinary systems for which ordinary bisimilarity is not a congruence, can gain this property (and a satisfactory algebraic structure) by also considering dynamic bisimilarity. This is done at the expense of a finer notion of observational congruence, which anyway is the coarsest possible, if it must be a bisimulation.

In a preliminary version of this paper which appeared as [7] similar concepts and results have been developed for the restricted class of rules in algebraic format [13], and the basic construction of the lifted functor considered only derived rules in DeSimone format [9]. Besides adding complete proofs, the present paper generalises these results to quite a wider class of SOS specifications, using in the definition of the lifted functor rules in TYFT format.

2. Structured operational semantics

Transition systems in this paper are always equipped with an algebraic structure determined by an equational algebraic specification. Thus we start reviewing some basic notions about algebras and algebraic specifications [10].

2.1. Preliminaries on algebraic specifications

We consider one-sorted specifications $\Gamma = \langle \Sigma, E\rangle$ where $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ is a family of operation symbols (we write $op : n$ for $op \in \Sigma_n$) and E is a (not necessarily finite) set of equations. A Σ-algebra $A = \langle |A|, (op^A)_{op\in\Sigma}\rangle$ consists of a carrier set $|A|$ and a family of operations such that $op^A : |A|^n \to |A|$ if $op : n \in \Sigma$. A Σ-homomorphism $f : A \to B$ is a function $f : |A| \to |B|$ between carriers which respects the operations, i.e., $op^B \circ f^n = f \circ op^A$. Denote the corresponding category by $\mathscr{A}lg(\Sigma)$.

Let T_Σ be the term algebra over Σ and, for a given set X of variables, $T_\Sigma(X)$ be the algebra of Σ-terms with variables in X. By $\mathscr{V}(t)$ we shall denote the set of variables actually occurring in a term (or any other syntactical entity) t. The term algebra T_Σ is an initial object in $\mathscr{A}lg(\Sigma)$. The unique homomorphism into a Σ-algebra A is given by the inductive evaluation of ground terms $eval : T_\Sigma \to A$. An *assignment* for a set of variables X into a Σ-algebra A is a function $v : X \to |A|$; if A is an algebra of terms, like $T_\Sigma(Y)$, such an assignment is also called a *substitution* and usually denoted with Greek letter (σ, ρ, ...). The term algebra $T_\Sigma(X)$ is free over X in $\mathscr{A}lg(\Sigma)$: if $v : X \to |A|$ is an assignment for X into A, its free extension is denoted by $\bar{v} : T_\Sigma(X) \to A$ (sometimes we shall denote it by $\bar{v}_A$, making explicit the target algebra A).

An equation (over X) is a pair of terms $s = t$ with $s, t \in T_\Sigma(X)$. It is satisfied in a Σ-algebra A if $\bar{v}(s) = \bar{v}(t)$ for each assignment $v : X \to |A|$. A Σ-algebra A satisfies an algebraic specification $\Gamma = \langle \Sigma, E\rangle$ if it satisfies all equations in E. In this case A is called Γ-algebra. The category of Γ-algebras and homomorphisms is the full subcategory $\mathscr{A}lg(\Gamma) \subseteq \mathscr{A}lg(\Sigma)$. The *forgetful functor* mapping a Γ-algebra to its carrier is denoted by $|\_| : \mathscr{A}lg(\Gamma) \to \mathbf{Set}$. Its left adjoint generating the free Γ-algebra over a set if $F^\Gamma : \mathbf{Set} \to \mathscr{A}lg(\Gamma)$.

Given a signature Σ, a *context* C over Σ is an element of $T_\Sigma(\{\bullet\})$ with exactly one occurrence of the variable $\bullet$. Given a Σ-algebra A and $a \in A$, by $C[a]$ we denote the element $\bar{v}(C)$ of A, where $v(\bullet) = a$. An equivalence relation $\mathscr{R} \subseteq |A| \times |A|$ is called *congruence* if it is preserved by the application of contexts, i.e., $\langle a, b\rangle \in \mathscr{R}$ implies $\langle C[a], C[b]\rangle \in \mathscr{R}$ for all $a, b \in |A|$ and every context C over Σ. An initial object in $\mathscr{A}lg(\Gamma)$ is constructed as the quotient of T_Σ w.r.t. the least congruence containing the relation $\{\langle \bar{v}(s), \bar{v}(t)\rangle \mid v : X \to |A| \wedge (s = t) \in E\}$ denoted by $T_\Sigma/_E$ or T_Γ.

2.2. Transition systems with algebraic structure

The basic idea of SOS specifications is to specify a transition relation by induction over the structure of the system's states. In order to make explicit this structure, instead of standard labelled transition systems we consider transition systems whose sets of states have an algebraic structure.

Definition 1 (*Heterogeneous transition systems*). Let $\Gamma = \langle \Sigma, E\rangle$ be an algebraic specification and L be a set of labels. *A* (*heterogeneous*)[3] *transition system over* Γ, L is a pair $hts = \langle A, \to_{hts}\rangle$ where A is a Γ-algebra and $\to_{hts} \subseteq |A| \times L \times |A|$ is a labelled (transition) relation. For $\langle a, l, b\rangle \in \to_{hts}$ we write $a \xrightarrow{l}_{hts} b$, as usual.

A morphism $f : hts \to hts'$ of (heterogeneous) transition systems over Γ and L is a Γ-homomorphism $f : A \to A'$ such that $a \xrightarrow{l}_{hts} b$ implies that $f(a) \xrightarrow{l}_{hts'} f(b)$. The category of (heterogeneous) transition systems over Γ and L is denoted $\mathbf{HTS}_{\Gamma,L}$.

A labelled transition system, *briefly LTS, is a heterogeneous transition system over the empty specification* $\Gamma = \emptyset$. *The category* $\mathbf{LTS}_L$ *of labelled transition systems* (*over* L) *is defined as* $\mathbf{HTS}_{\emptyset,L}$.

When considering systems with algebraic structure we will usually assume that the signature Σ contains at least one constant. This ensures that transition systems over Γ, L have non-empty carrier.

Bisimulation is usually defined for labelled transition systems. Below, this notion is lifted to heterogeneous systems by applying the classical definition to the labelled transition system obtained by forgetting the algebraic structure of states. Intuitively, two states of a labelled transition system are bisimilar if not only there are sequences of transitions starting from them having the same labels, but also the states reached after such transitions are bisimilar. *Bisimilarity* is the maximal set of pairs of bisimilar states, and it can be shown easily that it is a well-defined equivalence relation.

Definition 2 (*Bisimulation*). Let $\Gamma = \langle \Sigma, E\rangle$ be an algebraic specification, L be a set of labels, and $hts = \langle S, \to\rangle$ be a heterogeneous transition system over Γ, L. Let $\mathscr{R}$

[3] This qualification is intended to stress the fact that in these systems the labels and the transition relation have a weaker structure than the states, unlike *structured* transition systems introduced below.

be a binary relation on S. Then Ψ, a function from relations to relations, is defined by $(s,t) \in \Psi(\mathscr{R})$ if and only if for all $l \in L$:

- whenever $s \xrightarrow{l} s'$ there exists t' such that $t \xrightarrow{l} t'$ and $(s',t') \in \mathscr{R}$; and
- whenever $t \xrightarrow{l} t'$ there exists s' such that $s \xrightarrow{l} s'$ and $(s',t') \in \mathscr{R}$.

A relation $\mathscr{R}$ is called *bisimulation* if and only if $\mathscr{R} \subseteq \Psi(\mathscr{R})$.

The relation $\sim = \cup \{\mathscr{R} \mid \mathscr{R} \subseteq \Psi(\mathscr{R})\}$ is called *bisimilarity*.

On transition systems with algebraic structure on states, equivalences which are congruences with respect to the operators are very important: they can be used to provide a compositional abstract semantics. In many cases, bisimilarity is not a congruence, as we will see later on with an example for the π-calculus. This leads us naturally to the definition of *observational congruence*, which is simply the coarsest congruence included in bisimilarity.

Definition 3 (*Observational congruence*). Let $hts = \langle S, \rightarrow \rangle$ be a heterogeneous transition system over Γ and L, and $s,t \in S$ be two states of *hts*. We say that $s \approx t$ if and only if for any context C over Γ, $C[s] \sim C[t]$. Relation $\approx$ is called *observational congruence*.

Structured transition systems are systems where both the states and the transition relation are equipped with an algebraic structure, therefore they can be seen as heterogeneous transition systems over Γ and L where both L and the transition relation are Γ-algebras. A general theory of such systems has been proposed in [8], and has been used to provide a computational semantics for many formalisms (see Section 1). Next they are defined as labelled transition systems internal to a category of algebras.

Definition 4 (*Structured transition systems*). Let Γ be an algebraic specification and L be a Γ-algebra of labels. A *structured transition system* (*over Γ and L*) is a pair $sts = \langle A, \rightarrow_{sts} \rangle$ where A is a Γ-algebra of states and $\rightarrow_{sts} \subseteq A \times L \times A$ is a subalgebra of the product $A \times L \times A$ in $\mathscr{A}lg(\Gamma)$.

The category of structured transition systems over Γ and L, with morphisms defined as in Definition 1, is denoted $\mathbf{STS}_{\Gamma,L}$.

The concept of structured transition systems represents a generalisation of the algebraic semantics of place–transition (P/T) nets in [21]. There it is shown that the transition systems of P/T nets are naturally obtained by imposing a commutative monoid structure on the transitions of a *heterogeneous graph* representing a net.

2.3. SOS rules and specifications

Given an algebraic specification Γ and a set of labels L, a collection of SOS rules can be regarded as a specification of the subcategory of $\mathbf{HTS}_{\Gamma,L}$ including all transition systems for which the transition relation is closed under the given rules. In the following, SOS rules are formally defined as finite implications of sequents over a binary transition predicate $\_ \xrightarrow{l} \_$ for each label $l \in L$. Such rules may be interpreted as Horn clauses (with equality) specifying a heterogeneous transition system regarded as a relational structure.

Definition 5 (*SOS rules*, *satisfaction*, *entailment*, *theory*). Given a set of labels L, an algebraic specification $\Gamma = \langle \Sigma, E \rangle$, and a countable set of variables X, a sequent $s \xrightarrow{l} t$ (over L and Γ) is a triple where $l \in L$ is a label and $s, t \in T_\Sigma(X)$ are Σ-terms with variables in X. An *SOS rule* r over Γ, L, and X takes the form

$$\frac{s_1 \xrightarrow{l_1} t_1, \ldots, s_n \xrightarrow{l_n} t_n}{s \xrightarrow{l} t}$$

where $s_i \xrightarrow{l_i} t_i$ as well as $s \xrightarrow{l} t$ are sequents over Γ, L, and X.

Given a heterogeneous transition system $hts = \langle A, \rightarrow_{hts} \rangle$, an assignment $v : X \rightarrow |A|$ is a *solution* to a sequent $s \xrightarrow{l} t$ over Γ, L, and X in hts if $\bar{v}(s) \xrightarrow{l}_{hts} \bar{v}(t)$. We say that *hts satisfies* a rule r like above, written $hts \models r$, if each (joint) solution to $s_i \xrightarrow{l_i} t_i$ for $i = 1, \ldots, n$ is also a solution to $s \xrightarrow{l} t$. In this case we also say that *hts* is a *model* of r.

An *SOS specification* is a four-tuple $\Delta = \langle \Gamma, L, X, R \rangle$ consisting of an algebraic specification Γ, a set of labels L, a countable set of variables X, and a set of SOS rules R over Γ, L, and X. By $\mathbf{HTS}_\Delta$ we denote the full subcategory of $\mathbf{HTS}_{\Gamma,L}$ where all systems satisfy the rules in R.

An SOS specification Δ *entails* a rule r if all heterogeneous transition systems in $\mathbf{HTS}_\Delta$ also satisfy r. The theory $Th(\Delta)$ of Δ is defined as the closure of R under this entailment relation.

According to the formalisation of SOS rules has Horn clauses, a sequent $s \xrightarrow{l} t$ is a proposition stating that s and t are in the relation $\xrightarrow{l}$. Modulo this translations, the above notion of satisfaction of rules by transition systems coincides with the satisfaction of Horn clauses with equality by a corresponding relational structure. The following fact is a consequence of this observation.

Fact 6. *The category* $\mathbf{HTS}_\Delta$ *has an initial object* T_Δ *whose set of states is the initial* Γ*-algebra* T_Γ.

The next proposition shows that structured transition systems can be characterised, in quite an obvious way, by a suitable SOS specification.

Proposition 7 (Specifying structured transition systems). *Let* Γ *be a specification and* L *be a* Γ*-algebra of labels. Furthermore, let* X *be a countable set of variables, and let* R *consist of all rules*

$$[op] \quad \frac{x_1 \xrightarrow{l_1} y_1, \ldots, x_n \xrightarrow{l_n} y_n}{op(x_1, \ldots, x_n) \xrightarrow{op^L(l_1,\ldots,l_n)} op(y_1, \ldots, y_n)}$$

for each operation op of arity n in Γ, *and for any choice of labels* $l_1, \ldots, l_n \in |L|$. *Then, the category* $\mathbf{STS}_{\Gamma,L}$ *is isomorphic to the category* $\mathbf{HTS}_{\Delta}$ *with* $\Delta = \langle \Gamma, |L|, X, R \rangle$.

Rule [op] above shows that structured transition systems are only adequate for modelling rule-based systems (like Petri nets, term rewriting systems, etc.) where the algebraic structure is orthogonal to the transition structure. This is not the case, for example, for process algebras, and this is the reason why we introduced in Definition 1 systems where the structure of states does not necessarily carry over to transitions. Consider for example the following fragment of the π-calculus [24] with *early binding* (but without restriction or extrusion) which will be our main example.

Example 8 (π-*calculus fragment*). Assuming a countable infinite set $\mathcal{N}$ of *names* (ranged over by $x, y, z, \ldots$), the *prefixes* ($\alpha, \beta, \ldots$) are built according to the following syntax (we assume that $\tau \notin \mathcal{N}$):

$$\alpha = \tau \mid \bar{x}y \mid x(y).$$

Then *agents* are defined by the one-sorted algebraic specification $\Pi = \langle \Sigma_\Pi, E_\Pi \rangle$ whose signature is given by

$$P = 0 \mid \alpha . P \mid P + Q \mid P|Q \mid P[x/y]$$

where P, Q range over agents. Notice in particular that $\alpha.\_$ and $\_[x/y]$ both represent families of unary operations, one for each prefix α and each pair of names x, y, respectively. A simpler and more elegant presentation could have been given by using a *many-sorted* algebraic specification including, besides a sort for agents, also sorts for names and prefixes, and postulating a fixed interpretation for those additional sorts (in the style, for example, of Hidden Algebras [14]). We preferred to stick to the one-sorted case, to keep definitions simpler.

The equations below axiomatise the operation of substitution. They also make sure that agents are defined up to α-conversion, and that $\langle +, 0 \rangle$ forms a semi-lattice and $\langle |, 0 \rangle$ a commutative monoid. Notice that the equations involving assumptions about the names x, y, z, v are actually equation schemas expanding to a countable set of equations.

$$E_\Pi =$$

for all P, Q, R: *Agent*, $x, y, z, v \in \mathcal{N}$

$$0 + P = P, \qquad P + Q = Q + P,$$

$$P + P = P, \qquad (P + Q) + R = P + (Q + R),$$

$$(P|Q)|R = P|(Q|R), \quad P|Q = Q|P \quad P|0 = P,$$

$$0[z/x] = 0,$$

$$(\tau . P)[z/x] = \tau . P\,[z/x],$$

$$(\bar{x}y\,.\,P)[z/x] = \bar{z}y\,.\,P[z/x] \quad \text{if } x \neq y,$$

$$(\bar{x}y\,.\,P)[z/y] = \bar{x}z\,.\,P[z/y] \quad \text{if } x \neq y,$$

$$(\bar{x}y\,.\,P)[z/v] = \bar{x}y\,.\,P[z/v] \quad \text{if } v \notin \{x, y\},$$

$$(\bar{x}x\,.\,P)[z/x] = \bar{z}z\,.\,P[z/x],$$

$$x(y)\,.\,P = x(z)\,.\,P[z/y] \quad \text{if } z \notin \text{free-names}(P),$$

$$(x(y)\,.\,P)[z/v] = x(y)\,.\,P[z/v] \quad \text{if } y \notin \{z, v\} \text{ and } x \neq v,$$

$$(x(y)\,.\,P)[z/x] = z(y)\,.\,P[z/x] \quad \text{if } y \notin \{x, z\},$$

$$(P+Q)[z/x] = P[z/x] + Q[z/x],$$

$$(P|Q)[z/x] = P[z/x]|Q[z/x].$$

Let L_Π be the set of labels (observable actions) consisting of

output actions	$\bar{x}y$	for each $x, y \in \mathcal{N}$
input actions	xy	for each $x, y \in \mathcal{N}$
invisible action	τ	

The SOS specification **Pi** is given by the four-tuple $\mathbf{Pi} = \langle \Pi, L_\Pi, X, R_\Pi \rangle$, where R_Π consists of all instances of the following rules:

$$[out]\ \frac{}{\bar{x}y.P \xrightarrow{\bar{x}y} P} \qquad [in]\ \frac{}{x(y).P \xrightarrow{xz} P[z/y]} \quad \text{for each } z \in \mathcal{N}$$

$$[ch]\ \frac{P \xrightarrow{l} P'}{P+Q \xrightarrow{l} P'} \quad [par]\ \frac{P \xrightarrow{l} P'}{P|Q \xrightarrow{l} P'|Q} \quad [com]\ \frac{P \xrightarrow{\bar{x}y} P',\ Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\tau} P'|Q'}.$$

Due to the commutativity of $+$ and $|$ the symmetric variants of the last three rules are not needed, as they are entailed by the SOS specification.

The fragment of the π-calculus under consideration can be characterised as the initial model of this specification.

It is clear that the process algebra just introduced cannot be represented as a structured transition system because otherwise the transition relation would automatically be closed under all operations on states. This would mean to assume rules like in Proposition 7 which are clearly not meaningful here.

In fact, SOS rules provide a much more flexible way to specify the generation of transitions which is in general independent of the algebraic structure on states. However, we shall show in the next section that, under the coalgebraic view, models of *compositional* SOS specifications admit a presentation which is similarly homogeneous like structured transition systems. Unfortunately, as it is shown next, the SOS specification **Pi** does not generate a compositional transition system.

Example 9 (*The π-calculus is not compositional*). Let $u, v \in \mathcal{N}$ with $u \neq v$. In the initial **Pi**-transition system consider the two agents

$$P = \bar{u}y\,.\,0|v(z).0 \quad \text{and} \quad Q = \bar{u}y\,.\,v(z)\,.\,0 + v(z).\bar{u}y.0.$$

Clearly, $P \sim Q$. Now consider the context $C = x(v).\bullet$ over Π. Then it is easy to check that $C[P] = x(v).P \not\sim x(v).Q = C[Q]$. In fact, we have $x(v).P \xrightarrow{xu} P[u/v] = \bar{u}y\,.\,0|u(z).0 \xrightarrow{\tau} 0|0[y/z] = 0$, while $x(v).Q \xrightarrow{xu} Q[u/v] = \bar{u}y\,.\,u(z)\,.\,0 + u(z)\,.\,\bar{u}y.0$, and this last agent has no outgoing transitions labelled by τ.

Again, this is related to the fact that the transition relation is not compositional in the first place since the agent $P[u/v]$ can make a τ-transition which cannot be derived by the SOS rules from transitions out of P.

One may ask whether compositionality is a necessary condition for ensuring that bisimilarity is a congruence: the following example shows that this is not the case. Consider the algebraic structure given by a binary associative operation $\_ * \_$ together with a single constant a, and the SOS rule with empty premise and $a * a \xrightarrow{\alpha} a$ as conclusion. The initial system has as states all non-empty sequences of a's, and each transition reduces their length by one. Clearly, two states of this system are bisimilar if and only if they are equal. Still the behaviour is not compositional since the state $a * a$ may be decomposed, but there is no way to derive the outgoing transition from transitions out of a.

We conclude this section by recalling the well-known TYFT format of SOS rules [16], which shall play a central role in the rest of the paper.

Definition 10 (TYFT *and separated format of SOS rules*). An SOS rule r over $\Gamma = \langle \Sigma, E \rangle$ and L is in TYFT *format* if it has the form

$$r = \frac{\{t_i \xrightarrow{l_i} y_i\}_{i \in I}}{f(x_1, \ldots, x_n) \xrightarrow{l} t}$$

where y_i and x_j are all distinct variables. Rule r is *pure* if all its variables are among $\{y_i\}_{i \in I} \cup \{x_j\}_{1 \leqslant j \leqslant n}$. Furthermore, r is *look-ahead free* if for all $i \in I$, $\mathscr{V}(t_i) \subseteq \{x_1, \ldots, x_n\}$. Since we will only consider TYFT rules which are pure and look-ahead free, for the sake of conciseness in the following we will call them "TYFT rules" *tout court*. If R is a set of SOS rules, by TYFT$[R]$ we denote the set of all TYFT rules in R.

An SOS rule r is in *separated format* if it satisfies all the conditions for pure, look-ahead free TYFT rules, but for the fact that the source of the conclusion is an arbitrary term t with $\mathscr{V}(t) = \{x_1, \ldots, x_n\}$.

3. Coalgebraic models for heterogeneous transition systems

In this section we first review the representation of standard labelled transition systems as coalgebras for an endofunctor on the category of sets [28]. Then, we show how an SOS specification determines a lifting of this endofunctor to a category of algebras. Finally we discuss under which conditions a heterogeneous transition system can be represented as a coalgebra for this lifted functor thus emphasising the relevant algebraic structure of states and transitions. In particular we will show that this more structured presentation is feasible only for compositional systems.

3.1. Transition systems as coalgebras

Let us start introducing the formal definition of coalgebra for a functor.

Definition 11 (*Coalgebras*). Let $B:\mathscr{C}\to\mathscr{C}$ be an endofunctor on a category $\mathscr{C}$. A *coalgebra* for B or *B-coalgebra* is a pair $\langle A,a\rangle$ where A is an object of $\mathscr{C}$ and $a:A\to B(A)$ is an arrow. A *B-cohomomorphism* $f:\langle A,a\rangle\to\langle A',a'\rangle$ is an arrow $f:A\to A'$ of $\mathscr{C}$ such that

$$f;a' = a;B(f). \tag{1}$$

The category of B-coalgebras and B-cohomomorphisms will be denoted B-**Coalg**. The *underlying functor* $U:B\text{-}\mathbf{Coalg}\to\mathscr{C}$ maps an object $\langle A,a\rangle$ to A and an arrow f to itself.

For a fixed set of labels L, Let $Q_L:\mathbf{Set}\to\mathbf{Set}$ be the functor defined on objects as $X\mapsto\mathscr{P}(L\times X)$, where $\mathscr{P}$ denotes the powerset functor, and on arrows as $Q_L(f)(T)=\{\langle l,f(a)\rangle \mid \langle l,a\rangle\in T\}$, for $f:X\to Y$ and $T\subseteq L\times X$. Then coalgebras for this functor are one-to-one with labelled transition systems over L [28].

Proposition 12 (Labelled transition systems as coalgebras). *The category Q_L-**Coalg** is isomorphic to the sub-category of $\mathbf{LTS}_L$ containing all its objects and all the morphisms $f:TS\to TS'$ which also "reflect" transitions, i.e., such that if $f(s)\xrightarrow{l}_{TS'}t$ then there is a state $s'\in S$ such that $s\xrightarrow{l}_{TS}s'$ and $f(s')=t$.*

It is instructive to spell out the correspondence just stated. For objects, a transition system $\langle S,\to\rangle$ is mapped to the coalgebra $\langle S,\sigma\rangle$ where $\sigma(s)=\{\langle l,s'\rangle \mid s\xrightarrow{l}s'\}$, and, vice versa, a coalgebra $\langle S,\sigma:S\to Q_L(S)\rangle$ is mapped to the system $\langle S,\to\rangle$ with $s\xrightarrow{l}s'$ if $\langle l,s'\rangle\in\sigma(s)$. For arrows, by spelling out condition (1) for functor Q_L, we get

$$\forall s\in S.\{\langle l,t\rangle \mid f(s)\xrightarrow{l}t\}=\{\langle l,f(s')\rangle \mid s\xrightarrow{l}s'\}$$

and by splitting this set equality in the conjunction of the two inclusions, one can easily see that inclusion "$\supseteq$" is equivalent to $s\xrightarrow{l}s'\Rightarrow f(s)\xrightarrow{l}f(s')$, showing that f is a transition system morphism, while the left-to-right inclusion is equivalent to $f(s)\xrightarrow{l}t\Rightarrow\exists s'.s\xrightarrow{l}s'\wedge f(s')=t$, meaning that f is a "zig-zag" morphism, i.e., that it reflects transitions.

For technical reasons which will become clear in the next subsection, we shall not use functor Q_L to represent transition systems over L, but the slightly different functor $P_L : \mathbf{Set} \to \mathbf{Set}$ defined on objects as $P_L(X) = \mathscr{P}(L \times X + X)$,[4] and on arrows as $P_L(f)(T) = \{\langle l, f(a)\rangle \mid \langle l,a\rangle \in T\} \cup \{f(a) \mid a \in T\}$, for $f : X \to Y$ and $T \subseteq L \times X + X$. More precisely, a transition system $\langle S, \to\rangle$ is represented by the P_L-coalgebra $\langle S, \sigma : S \to \mathscr{P}(L \times S + S)\rangle$ defined as $\sigma(s) = \{\langle l, s'\rangle \mid s \xrightarrow{l} s'\} \cup \{s\}$. Therefore the only difference with the representation of S as a Q_L-coalgebra is that the set of transitions associated to a state s includes the state itself. By the way, this defines a full embedding of Q_L-**Coalg** into P_L-**Coalg**, which guarantees that the interpretation of cohomomorphisms as functions preserving and reflecting transitions holds for the representation of systems as P_L-coalgebras as well.

The property of "reflecting behaviours" enjoyed by cohomomorphims plays a fundamental rôle, for example, for the characterisation of bisimulation relations as spans of cohomomorphisms, for the relevance of final coalgebras, and for various other results of the theory of coalgebras [28]. Given two coalgebras $\langle A, a\rangle$ and $\langle A', a'\rangle$, a *coalgebraic bisimulation* on them is a coalgebra $\langle A \times A', r\rangle$ having as carrier the cartesian product of the carriers, and such that the projections $\pi : A \times A' \to A$ and $\pi' : A \times A' \to A'$ are cohomomorphisms. Interestingly, it is easy to check that two states of a labelled transition system S are bisimilar (in the standard sense, see Definition 2) if and only if there is a coalgebraic bisimulation on S (regarded as a Q_L- or P_L-coalgebra) which relates them.

An even easier definition of categorical bisimilarity can be given if there exists a final coalgebra. In this case, two elements of the carrier of a coalgebra are bisimilar iff they are mapped to the same element of the final coalgebra by the unique cohomomorphism. Unfortunately, due to cardinality reasons, the functor P_L does not admit a final coalgebra (the same holds for Q_L [28]). One satisfactory, alternative solution consists in replacing the powerset functor $\mathscr{P}$ on **Set** by the *countable* powerset functor $\mathscr{P}_c$, which maps a set to the family of its countable subsets. Then, defining the functor $P_L^c : \mathbf{Set} \to \mathbf{Set}$ by $X \mapsto \mathscr{P}_c(L \times X + X)$, coalgebras for this endofunctor allow one to represent transition systems with *countable degree*, i.e., systems where for each state $s \in S$ the set $\{\langle s', l\rangle \mid s \xrightarrow{l} s'\} \cup \{s\}$ is countable. Unlike the functor P_L, the functor P_L^c admits cofree and final coalgebras.

Proposition 13 (Final and cofree P_L^c-coalgebras). *The obvious underlying functor U : P_L^c-**Coalg** $\to$ **Set** has a right adjoint $R : \mathbf{Set} \to P_L^c$-**Coalg** associating with each set X a cofree coalgebra over X. As a consequence, the category P_L^c-**Coalg** has a final object, which is the cofree coalgebra $R(\mathbf{1})$ over a final set* **1**.

Proof. According to [28] it is enough to show that the functor P_L^c is bounded. This is the case, because the cardinality of the subsets assigned by P_L^c is bounded by ω. □

We shall stick to this functor throughout the rest of the paper, and since there is no room for confusion the superscript c will be understood.

[4] With $+$ we denote disjoint union. Therefore an element $T \in P_L(X)$ may contain pairs like $\langle l,x\rangle \in L \times X$ and/or elements of X.

3.2. Lifting functors to categories of algebras

The transition systems in this paper are equipped with some algebraic structure on states, transitions, and/or labels which plays relevant role in their construction and analysis. Therefore, their representation as coalgebras in **Set** introduced above is not satisfactory because the algebraic structure on states (and transitions) is lost. This calls for the introduction of *structured coalgebras*, i.e., coalgebras for an endofunctor on a category $\mathscr{A}lg(\Gamma)$ of algebras for an algebraic specification Γ which is determined by the structure of states. Since it is natural to require that the structured coalgebraic representation of a system is compatible with the unstructured, set-based one, the following notion will be relevant.

Definition 14 (*Lifting*). Given endofunctors $B:\mathscr{C}\to\mathscr{C}$, $B':\mathscr{C}'\to\mathscr{C}'$ and a functor $V:\mathscr{C}'\to\mathscr{C}$, B' is called a *lifting of B along V*, if $B';V=V;B$.

In particular, if $|\_|:\mathscr{A}lg(\Gamma)\to\mathbf{Set}$ is the underlying set functor, for a given SOS specification $\Delta=\langle\Gamma,L,X,R\rangle$ satisfying certain conditions we will define a functor $P^\Delta:\mathscr{A}lg(\Gamma)\to\mathscr{A}lg(\Gamma)$ which is a lifting of P_L along $|\_|$.

The structured coalgebraic representation of transition systems has been studied in [29] for the case of CCS and other process algebra whose operational semantics are given by SOS rules in the GSOS format, and in [5] for structured transition systems. In the first case the lifting of P_L is determined by the SOS rules, while in the second one it is induced by the algebraic specification Γ. We follow the first approach and show subsequently how the second forms a special case. Given a Γ-algebra A with $\Gamma=\langle\Sigma,E\rangle$, the definition of lifting uniquely determines the carrier of $P^\Delta(A)$, because $|P^\Delta(A)|$ must be equal to $P_L(|A|)=\mathscr{P}(L\times|A|+|A|)$. Therefore to define the action of the lifted functor P^Δ on A, we only have to provide the interpretation of all operator symbols in Σ on the carrier $\mathscr{P}(L\times|A|+|A|)$.

Since the carrier of $P^\Delta(A)$ is a power-set, what we need to define is a *power-algebra* [12]. The elements of such an algebra are sets containing pairs of the form $\langle l,a\rangle$ and elements of A. Actually, we are mainly interested in sets which contain only one element of A, and we interpret them as the possible transitions out of that state. The interpretation of the operators of Σ is driven by the SOS rules: each TYFT rule having a conclusion of the shape $f(x_1,\dots,x_n)\xrightarrow{l}t$ gives a contribution to the definition of f on the power-algebra. Intuitively, if there are transitions out of the states $\bar v(x_1),\dots,\bar v(x_n)$ which satisfy the premises (where $v:X\to|A|$ provides an assignment of the variables of the rule to elements of $|A|$), the application of f to sets containing such transitions will return a set of transitions out of the composed state containing at least the transition $\langle l,\bar v(t)\rangle$.

Definition 15 (*Power algebras*). Let $\Gamma=\langle\Sigma,E\rangle$ be an algebraic specification. For a Γ-algebra A, we will denote by $PowAlg(\Gamma,L,A)$ the class of power-algebras containing all Γ-algebras P such that:

(1) the carrier of P is $\mathscr{P}(L\times|A|+|A|)$;

(2) all operations are monotonic, i.e., for all $f \in \Sigma_n$, if $T_i \subseteq T_i'$ for all $i \in \{1,\dots,n\}$, then $f^P(T_1,\dots,T_n) \subseteq f^P(T_1',\dots,T_n')$;
(3) for all $f \in \Sigma_n$ and for all $T_1,\dots,T_n \in |P|$, $f^P(T_1,\dots,T_N) \cap |A| = \{f^A(a_1,\dots,a_n) \mid \forall j \in \{1,\dots,n\} . a_j \in (T_j \cap |A|)\}$.[5]

We will denote by $|P_A|$ the set $\mathscr{P}(L \times |A| + |A|)$, i.e., the carrier of all algebras in $PowAlg(\Gamma, L, A)$.

Let P and Q be two power-algebras in $PowAlg(\Gamma, L, A)$. We write $P \sqsubseteq Q$ if for all $n \in \mathbb{N}$, for all operators $f \in \Sigma_n$ and for all n-tuples $\langle T_1,\dots,T_n\rangle \in |P_A|^n$ it holds $f^P(T_1,\dots,T_n) \subseteq f^Q(T_1,\dots,T_n)$.[6]

Given *a* ω-chain $P_0 \sqsubseteq P_1 \sqsubseteq \dots$ of algebras in $PowAlg(\Gamma, L, A)$, it admits a limit P with $P \in PowAlg(\Gamma, L, A)$, where the action of an operation on a tuple of sets is defined as the union of the actions of the corresponding operations in the chain.

The following technical lemma will be used later.

Fact 16 (Properties of power-algebras). (1) *Let $P \in PowAlg(\Gamma, L, A)$ be a power-algebra, $t \in T_\Sigma(X)$ be a term, and $v : \mathscr{V}(t) \to |P_A|$ be an assignment. Then $a \in \bar{v}_P(t) \cap |A| \Leftrightarrow \exists w : \mathscr{V}(t) \to |A| . \forall x \in \mathscr{V}(t) . w(x) \in v(x) \wedge a = \bar{w}_A(t)$.*

(2) *Let $P, Q \in PowAlg(\Gamma, L, A)$ and $v : X \to |P_A|$. Then for all $t \in T_\Sigma(X)$, $\bar{v}_P(t) \cap |A| = \bar{v}_Q(t) \cap |A|$.*

Proof (Sketch). (1) is easily proved by structural induction on t, and (2) follows from point (1), by observing that in the formula characterizing $a \in \bar{v}_P(t)$, P does not appear at all. $\square$

We will define P^Δ, the lifting of P_L determined by a given SOS specification Δ, as the functor mapping a Γ-algebra A to the "minimal" power-algebra in $PowAlg(\Gamma, L, A)$ satisfying the TYFT rules in the theory of Δ.

Definition 17 (*Satisfaction of sequents and rules*). Let $P \in PowAlg(\Gamma, L, A)$, let $s \xrightarrow{l} t$ be a sequent (over Γ, L and X), and let $v : X \to |P_A|$ be an assignment. Then $\langle P, v\rangle$ is a *solution* to the sequent (written $\langle P, v\rangle \models s \xrightarrow{l} t$) if for all $a \in \bar{v}_P(t) \cap |A|$ it holds $\langle l, a\rangle \in \bar{v}_P(s)$.

Let r be a TYFT rule (over Γ, L, and X)

$$r = \frac{\{t_i \xrightarrow{l_i} y_i\}_{i \in I}}{f(x_1,\dots,x_n) \xrightarrow{l} t}$$

and let $v : \mathscr{V}(r) \to |P_A|$ be an assignment. We say that v is Y-*linear* if for all $i \in I$, $v(y_i) \cap |A|$ is a singleton. The pair $\langle P, v\rangle$ *satisfies* r (written $\langle P, v\rangle \models r$) if either v is not Y-linear, or it is Y-linear and $(\forall i \in I . \langle P, v\rangle \models t_i \xrightarrow{l_i} y_i) \Rightarrow (\langle P, v\rangle \models f(x_1,\dots,x_n) \xrightarrow{l} t)$.

[5] Let $\mathscr{P}(A)$ denote the power algebra obtained by extending the operations of A in a pointwise manner to subsets of $|A|$. Then this condition is equivalent to require that $\_ \cap |A| : P \to \mathscr{P}(A)$ is a Γ-homomorphism.

[6] Equivalently, $P \sqsubseteq Q$ iff for all $t \in T_\Sigma(X)$, for all $v : \mathscr{V}(t) \to |P_A|$, $\bar{v}_P(t) \subseteq \bar{v}_Q(t)$.

Similarly, we write $P \models s \xrightarrow{l} t$ ($P \models r$) if for all assignments $v : X \to |P|$ it holds $\langle P, v\rangle \models s \xrightarrow{l} t$ ($\langle P, v\rangle \models r$, respectively). If R is a set of sequents or rules, we write $P \models R$ if $P \models r$ for all $r \in R$.

In the definition of satisfaction for a TYFT rule by an assignment the restriction to Y-linear assignments only is motivated by the fact that the conclusion must be satisfied when exactly one transition is provided as witness for each premise of the rule: a non-Y-linear assignment v satisfies at least one premise either with no transition at all (if $v(y_i) \cap |A| = \emptyset$) or with more than one transition.

The definition of solution to a sequent above should make clear why we are considering the lifting of functor P_L rather than of Q_L (see Section 3.1). In order to provide a solution for a sequent $x \xrightarrow{l} y$ over X, an assignment $v : X \to |P_A|$ should map y to elements of $|A|$ and x to pairs representing transitions: this explains why the carrier of the power-algebras we are considering contains a copy of $|A|$.

Before introducing the lifting of P_L we need the following definition.

Definition 18 (TYFT *proof*). Let R be a set of SOS rules, let $s, t \in T_\Sigma(X)$, and let $\Phi = \{v_j \xrightarrow{m_j} z_j\}_{j\in J}$ be a finite set of sequents such that $v_j \in \mathscr{V}(s)$ for all $j \in J$, all elements of $Z = \{z_j \mid j \in J\}$ are distinct variables, $Z \cap \mathscr{V}(s) = \emptyset$, and $\mathscr{V}(t) \subseteq \mathscr{V}(\Phi)$ $[= \mathscr{V}(s) \cup Z]$.

Then we say that there is a TYFT *proof* (using R) of the sequent $s \xrightarrow{l} t$ with premises Φ if either

- $s \xrightarrow{l} t \in \Phi$, and in this case the TYFT proof is called *trivial*; or
- there is a TYFT rule $r = \dfrac{\{t_i \xrightarrow{l_i} y_i\}_{i\in I}}{f(x_1,\ldots,x_n) \xrightarrow{l} t'}$ in R and a substitution $\sigma : \{x_1, \ldots, x_n\} \to T_\Sigma(\mathscr{V}(s))$ such that
  (1) $\bar\sigma(f(x_1, \ldots, x_n)) = s$;
  (2) for all $i \in I$ there is a TYFT proof of a sequent $\bar\sigma(t_i) \xrightarrow{l_i} p_i$ with premises $\Phi_i = \{v \xrightarrow{m} z \in \Phi \mid v \in \mathscr{V}(\bar\sigma(t_i))\}$;
  (3) it holds $t = \bar\rho(t')$, where $\rho : \mathscr{V}(r) \to T_\Sigma(\mathscr{V}(\Phi))$ is the substitution defined as

$$\rho(x) = \begin{cases} \sigma(x) & \text{if } x \in \{x_1, \ldots, x_n\} \\ p_i & \text{if } x = y_i \wedge i \in I \end{cases}.$$

Definition 19 (*Lifting of P_L induced by a SOS specification*). Let $\Delta = \langle \Gamma = \langle \Sigma, E\rangle, L, X, R\rangle$ be a SOS specification such that for all equations $s = t$ in E the following property holds:

TYFT ***bisimilarity***: For every TYFT *proof* using $\mathrm{Th}(\Delta)$ of a sequent $\bar v(s) \xrightarrow{l} s'$, with $v : \mathscr{V}(s) \cup \mathscr{V}(t) \to T_\Sigma(X)$ and premises ϕ, there is a TYFT proof using exactly the same premises ϕ of a sequent $\bar v(t) \xrightarrow{l} t'$, for some t' such that $s' \equiv_E t'$, and vice versa.[7]

[7] By $\equiv_E$ we denote the congruence of $T_\Sigma(X)$ generated by the equations E of the SOS specification.

Then for a Γ-algebra A, $\mathscr{P}^{\Delta}(A)$ is defined as the minimal power-algebra $P_A^{\Delta} \in PowAlg(\Gamma, L, A)$ such that $P_A^{\Delta} \models r$ for all rules r in $Th(\Delta)$ which are in TYFT format. Furthermore, if $g : A \to B$ is a Γ-homomorphism, $\mathscr{P}^{\Delta}(g) : \mathscr{P}^{\Delta}(A) \to \mathscr{P}^{\Delta}(B)$ is defined as $\mathscr{P}^{\Delta}(g)(T) = \{g(a) \mid a \in T \cap |A|\} \cup \{\langle l, g(a)\rangle \mid \langle l, a\rangle \in T \cap (L \times |A|)\}$.

Notice that in the above definition only the rules in TYFT format belonging to the theory of the SOS specification are taken into account. The rest of this section is dedicated to the proof that, under the required condition on equations, P^{Δ} is a well-defined functor. Let us start by characterising in a constructive way the minimal power-algebra satisfying a given set of TYFT rules.

Proposition 20 (A chain of power-algebras). *Let $\Delta = \langle \Gamma, L, X, R\rangle$ be a SOS specification, and let $A \in \mathscr{A}lg(\Gamma)$. The power-algebras P_k^A, $k \in \mathbb{N}$ are defined inductively as follows*:

Base case: *For each $n \in \mathbb{N}$, $f \in \Sigma_n$, and $T_1, \ldots, T_n \in \mathscr{P}(L \times |A| + |A|)$,*

$$f^{P_0^A}(T_1, \ldots, T_n) = \{f^A(a_1, \ldots, a_n) \mid \forall j \in \{1, \ldots, n\}. a_j \in T_j \cap |A|\}.$$

Inductive case: *For each $k \geqslant 0$, the operations of P_{k+1}^A are defined as*

$$f^{P_{k+1}^A}(T_1, \ldots, T_n) = \begin{cases} f^{P_k^A}(T_1, \ldots, T_n) \cup \\ \{\langle l, a\rangle \mid \exists r \in \text{TYFT}[Th(\Delta)],\ r = \dfrac{\{t_i \xrightarrow{l_i} y_i\}_{i \in I}}{f(x_1, \ldots, x_n) \xrightarrow{l} t} \\ \exists v : \mathscr{V}(r) \to |P_A| . v \text{ is Y-linear} \wedge \forall j \in \{1, \ldots, n\}. v(x_j) = T_j \wedge \\ \langle P_k^A, v\rangle \models \{t_i \xrightarrow{l_i} y_i\}_{i \in I} \wedge a \in \bar{v}_{\mathscr{P}_k^A}(t) \cap |A|\} \end{cases}.$$

Then the following facts hold:

(1) *For all $k \in \mathbb{N}$, $P_k^A \in PowAlg(\Sigma, L, A)$.* [8]

(2) *For all $k \in \mathbb{N}$, $P_k^A \sqsubseteq P_{k+1}^A$.*

(3) *Let $P_A \in PowAlg(\Sigma, L, A)$ be the limit of the chain $P_0^A \sqsubseteq P_1^A \sqsubseteq \ldots$. Then P_A is the minimal power-algebra in $PowAlg(\Sigma, L, A)$ which satisfies all rules in* TYFT$[Th(\Delta)]$, *i.e.*,

 (3.1) $P_A \in PowAlg(\Sigma, L, A)$,

 (3.2) $P_A \models \text{TYFT}[Th(\Delta)]$, *and*

 (3.3) $Q \models \text{TYFT}[Th(\Delta)] \Rightarrow P_A \sqsubseteq Q$.

Proof. For the sake of simplicity, for an assignment $v : X \to |P_A|$ let us denote by $\bar{v}_k$ the induced function $\bar{v}_{P_k^A} : T_{\Sigma}(X) \to P_k^A$.

For point (1), the monotonicity of $f^{P_k^A}$ follows by the observation that if a Y-linear assignment $v : \mathscr{V}(r) \to |P_A|$ is a solution to the premises of r, then so is any v' such that $\forall i \in I . v'(y_i) = v(y_i) \wedge \forall j \in \{1, \ldots, n\} . v'(x_j) \supseteq v(x_j)$. Condition (3) of Definition 15

[8] Note that in this proposition the power-algebras P_k^A are regarded as Σ-algebras, instead of as Γ-algebras.

is obvious for $k=0$, and it is invariant for $k>0$ because no elements of $|A|$ are added in the Inductive Case.

Point (2) follows directly from the definitions, the only thing to be checked being that for a given rule r and Y-linear assignment v, if $\langle P_k^A, v\rangle$ is a solution to the premises of r, then so is $\langle P_{k+1}^A, v\rangle$. For a given $i\in I$, by Definition 17 $\langle P_k^A, v\rangle \models t_i \xrightarrow{l_i} y_i$ iff $\forall a\in \bar{v}_k(y_i)\cap |A|.\langle l_i, a\rangle \in \bar{v}_k(t_i)$. Since by Fact 16(2) we have $\bar{v}_k(y_i)\cap|A| = \bar{v}_{k+1}(y_i)\cap |A|$, and $\bar{v}_k(t_i)\subseteq \bar{v}_{k+1}(t_i)$ by easy structural induction, we deduce that $\langle P_{k+1}^A, v\rangle \models t_i \xrightarrow{l_i} y_i$ as desired.

Point (3.1) immediately follows from the fact that $PowAlg(\Sigma, L, A)$ is closed under limits of chains. Concerning point (3.2), suppose that there is a rule $r\in Th(\Delta)$ with

$$r = \frac{\{t_i \xrightarrow{l_i} y_i\}_{i\in I}}{f(x_1,\ldots,x_n)\xrightarrow{l} t}$$

and a Y-linear assignment $v:\mathscr{V}(r)\to |P_A|$ such that $\forall i\in I.\langle P_A, v\rangle \models t_i \xrightarrow{l_i} y_i$. Since P_A is defined as the limit of a chain of power-algebras there is a $k\in\mathbb{N}$ such that $\forall i\in I$. $\langle P_k^A, v\rangle \models t_i \xrightarrow{l_i} y_i$. By the Inductive Case above, this implies that $\forall a \in \bar{v}_k(t)\cap|A|.\langle l, a\rangle \in f^{P_{k+1}^A}(v(x_1),\ldots,v(x_n)) = \bar{v}_{k+1}(f(x_1,\ldots,x_n))$, which in turn implies $\langle P_A, v\rangle \models r$ because $\bar{v}_{k+1}(t)\subseteq \bar{v}_{P_A}(t)$. Finally, point (3.3) can be proved in a standard way by induction. □

Notice that, so far, for a given Γ algebra A we only know that $P^\Delta(A)$ is a Σ-algebra. The next example shows that P^Δ does not preserve, in general, the satisfaction of equations if the TYFT *bisimilarity* condition of Definition 19 is not satisfied.

Example 21 (*Failure of preservation of equations by P^Δ*). Consider the SOS specification $\Delta = \langle\langle \Sigma, E\rangle, L, X, R\rangle$ where Σ contains a single operator c of arity zero, $E=\{x=y\}$, $L=\{l\}$, and $X\supseteq\{x,y\}$. Clearly, each $\langle\Sigma,E\rangle$-algebra is isomorphic to $A=\langle\{*\}, c^A = *\rangle$. Since P^Δ is a lifting of P_L we have that $|P^\Delta(A)| = P_L(\{*\}) = \mathscr{P}(\{l\}\times\{*\}+\{*\}) = \{\emptyset, \{*\}, \{\langle l,*\rangle\}, \{*, \langle l, *\rangle\}\}$. No algebra defined on this carrier satisfies the equation $x=y$.

Notice that this counterexample is independent of the rules R. In particular, it is possible to show that equation $x=y$ does not satisfy the condition TYFT *bisimilarity* for any set of rules R. In fact, there is a (*trivial*) TYFT proof for sequent $x\xrightarrow{l} z$ with premise $\{x\xrightarrow{l} z\}$, but there is no TYFT proof for $y\xrightarrow{l} z$ using the same premise.

The following lemma relates the structure of the power-algebra $P^\Delta(A) = P_A$ to the existence of TYFT proofs for sequents.

Lemma 22 (on the structure of $P^\Delta(A)$). *Let $s\in T_\Sigma(X)$ and let $w : X\to |P_A|$ be an assignment. Furthermore, let Z be the set of variables $Z=\{z_{v,l,b} \mid \langle l,b\rangle \in w(v) \wedge v\in \mathscr{V}(s)\}$, let Φ_w be the set of sequents $\Phi_w = \{v\xrightarrow{l} z_{v,l,b} | z_{v,l,b}\in Z\}$, and let $w^Z : \mathscr{V}(s)\cup Z\to |P_A|$ be the assignment such that $w^Z(v) = w(v)$ for all $v\in\mathscr{V}(s)$, and $w^Z(z_{v,l,b}) = \{b\}$ for all $z_{v,l,b}\in Z$.*

Then the following two facts are equivalent:
(1) $\langle l,b\rangle \in \bar{w}(s)$;
(2) *there is a* TYFT *proof of a sequent* $s \xrightarrow{l} t$ *with premises* Φ_w, *and* $b \in \overline{w^Z}(t) \cap |A|$.

Proof (2) ⇒ (1). We proceed by induction on the depth of the TYFT proof for the sequent $s \xrightarrow{l} t$.

Base Case. If the TYFT proof is trivial, we have $s \xrightarrow{l} t \in \Phi_w$. By the definition of Φ_w and Z, this holds if and only if s is a variable, $t = z_{s,l,b}$ for some $b \in |A|$, and $\langle l,b\rangle \in w(s) = \bar{w}(s)$. Then the statement follows by observing that $\overline{w^Z}(t) = w^Z(z_{s,l,b}) = \{b\}$ by definition.

Inductive Case. Let us assume that the conditions listed in the second item of Definition 18 hold literally. By applying the induction hypothesis to the TYFT proofs of sequents $\bar{\sigma}(t_i) \xrightarrow{l_i} p_i$ (see point (2) there), we have that $\langle l_i,b_i\rangle \in \bar{w}(\bar{\sigma}(t_i))$, for $b_i \in \overline{w^Z}(p_i)$. This implies that $\langle P_A, w^Z \circ \rho\rangle \models \{t_i \xrightarrow{l_i} y_i\}_{i\in I}$ (where ρ is as in point (3) of Definition 18), and thus $\langle P_A, w^Z \circ \rho\rangle \models f(x_1,\dots,x_n) \xrightarrow{l} t'$, because $P_A \models r$. Thus for all $b \in \overline{w^Z}(\bar{\rho}(t')) = \overline{w^Z}(t)$ we have $\langle l,b\rangle \in \overline{w^Z}(\bar{\rho}(f(x_1,\dots,x_n))) = \overline{w^Z}(s) = \bar{w}(s)$, which concludes this part of the proof.

(1) ⇒ (2). Since by Proposition 20 P_A is the limit of the chain $\{P_k^A\}_{k\in\mathbb{N}}$, $\langle l,b\rangle \in \bar{w}(s)$ for $w : \mathscr{V}(s) \to |P_A|$ implies that there is a $k \in \mathbb{N}$ such that $\langle l,b\rangle \in \overline{w_{P_k^A}}(s)$, where $\overline{w_{P_k^A}}$ is the free extension of assignment w relative to algebra P_k^A. We show that point (2) of the statement holds using induction on both k and the structure of s.

Base Case. Suppose that $\langle l,b\rangle \in \overline{w_{P_0^A}}(s)$. By the definition of P_0^A in the *Base Case* of Proposition 20, this is only possible if s is a variable. In this case, from the hypothesis $\langle l,b\rangle \in \overline{w_{P_0^A}}(s) = w(s)$ we immediately deduce that $z_{s,l,b} \in Z$, $s \xrightarrow{l} z_{s,l,b} \in \Phi_w$ and $w^Z(z_{s,l,b}) = \{b\}$ by the above definitions. Therefore there is a trivial TYFT proof for sequent $s \xrightarrow{l} z_{s,l,b}$ with premises Φ_w, and $b \in \overline{w^Z}(z_{s,l,b})$, as required.

Inductive Case. Suppose now that $\langle l,b\rangle \in \overline{w_{P_{k+1}^A}}(s)$, and, without loss of generality, that s is not a variable (otherwise also $\langle l,b\rangle \in \overline{w_{P_0^A}}(s)$ holds, and the *Base Case* applies). Assuming that $s = f(s_1,\dots,s_n)$, we have $\langle l,b\rangle \in f^{P_{k+1}^A}(\overline{w_{P_{k+1}^A}}(s_1),\dots,\overline{w_{P_{k+1}^A}}(s_n)) \subseteq f^{P_{k+1}^A}(\bar{w}(s_1),\dots,\bar{w}(s_n))$.

As induction hypotheses, we assume that the statement holds true (IH1) for $\langle l',b'\rangle \in \bar{w}(s')$, if s' is a proper subterm of s, as well as (IH2) for $\langle l',b'\rangle \in g^{P_k^A}(\bar{w}(s'_1),\dots,\bar{w}(s'_m))$, with $g : m \in \Sigma$ and $s'_1,\dots,s'_m \in T_\Sigma(\mathscr{V}(s))$.

By Proposition 20, there is a rule $r = \frac{\{t_i \xrightarrow{l_i} y_i\}_{i\in I}}{f(x_1,\dots,x_n) \xrightarrow{l} t'}$ in TYFT$[Th(\Delta)]$ and a Y-linear assignment $u : \mathscr{V}(r) \to |P_A|$ such that
(a) $\forall j \in \{1,\dots,n\}\,.\,u(x_j) = \bar{w}(s_j)\ \wedge$
(b) $\langle P_k^A, u\rangle \models \{t_i \xrightarrow{l_i} y_i\}_{i\in I}$ $[\Leftrightarrow\ \exists b_i\,.\,u(y_i) \cap |A| = \{b_i\} \wedge \langle l_i,b_i\rangle \in \overline{u_{P_k^A}}(t_i)]\ \wedge$
(c) $b \in \overline{u_{P_k^A}}(t') \cap |A|$.

We show now how to construct a term $t \in T_\Sigma(\mathscr{V}(\Phi_w))$ and a TYFT proof of sequent $s \xrightarrow{l} t$ by exploiting rule r and the induction hypotheses. Let $\sigma : \{x_1,\dots,x_n\} \to$

$T_\Sigma(\mathscr{V}(s))$ be the substitution defined as $\sigma(x_j)=s_j$ for $j \in \{1,\dots,n\}$: by point (a) we have $u(x_j)=\bar{w}(\sigma(x_j))$, and thus $\overline{u_{P_k^\Delta}}(t) \subseteq \bar{w}(\bar{\sigma}(t))$ for any term $t \in T_\Sigma(\mathscr{V}(s))$. We apply the induction hypotheses to the premises of the rule, namely to the memberships $\langle l_i, b_i\rangle \in \overline{u_{P_k^\Delta}}(t_i) \subseteq \bar{w}(\bar{\sigma}(t_i))$ for all $i \in I$. We need to distinguish two cases. If t_i is a variable in $\{x_1,\dots,x_n\}$, say $t_i = x_j$, then $\overline{u_{P_k^\Delta}}(t_i) = u(x_j) = \bar{w}(s_j)$, thus (IH1) applies. If t_i is not a variable, let $t_i = g(t'_1,\dots,t'_m)$ for some $g : m \in \Sigma$. Then $\overline{u_{P_k^\Delta}}(t_i) = g^{P_k^\Delta}(\overline{u_{P_k^\Delta}}(t'_1),\dots,\overline{u_{P_k^\Delta}}(t'_m)) \subseteq g^{P_k^\Delta}(\bar{w}(\bar{\sigma}(t'_1)),\dots,\bar{w}(\bar{\sigma}(t'_m)))$, and in this case (IH2) applies. In both cases, for each $i \in I$ there is TYFT proof of a sequent $\bar{\sigma}(t_i) \xrightarrow{l_i} p_i$ using as premises $\{v \xrightarrow{l} z \in \Phi_w \mid v \in \mathscr{V}(\bar{\sigma}(t_i))\}$, and such that (†) $b_i \in \overline{w^Z}(p_i) \cap |A|$.

Now let $\rho : \mathscr{V}(r) \to \mathscr{V}(\Phi_w)$ be defined as in point (3) of Definition 18. Since $u(y_i) \cap |A| = \{b_i\}$ for all $i \in I$, by (†) above we infer that $u(y) \cap |A| \subseteq \overline{w^Z}(\rho(y)) \cap |A|$ for all $y \in \mathscr{V}(r)$. Finally, let t be defined as $t = \bar{\rho}(t')$. In this way, by construction we have a TYFT proof of sequent $s \xrightarrow{l} t$ with premises Φ_w, and $b \in \overline{u_{P_k^\Delta}}(t') \cap |A| \subseteq \overline{w^Z_{P_k^\Delta}}(\bar{\rho}(t')) \cap |A| = \overline{w^Z_{P_k^\Delta}}(t) \cap |A| \subseteq \overline{w^Z}(t) \cap |A|$, and this concludes the proof.

If all equations in E satisfy the TYFT bisimilarity condition then $P^\Delta(A)$ is a Γ-algebra if so is A.

Proposition 23 (Preservation of equations by P^Δ). *Consider an SOS specification $\Delta = \langle \Gamma = \langle \Sigma, E\rangle, L, X, R\rangle$ such that all equations $s = t$ in E satisfy the* TYFT *bisimilarity condition of Definition* 19. *Then for all $A \in \mathscr{A}lg(\Gamma)$ we have $P^\Delta(A) \in \mathscr{A}lg(\Gamma)$.*

Proof (*Sketch*). We have to show that $P^\Delta(A) \stackrel{def}{=} P_A$ satisfies all equations in E. Let $s = t$ be an equation over X in E and let $v : X \to \mathscr{P}(L \times |A| + |A|)$ be an assignment. Firstly, the fact that $\bar{v}_{P_A}(s) \cap |A| = \bar{v}_{P_A}(t) \cap |A|$ easily follows from Fact 16. Now assume that a value $\langle l, b\rangle$ is in $\bar{v}_{P_A}(s)$. Then there is a sequent $s \xrightarrow{l} s'$ and a TYFT proof for it with premises Φ_v and $b \in \overline{v^Z}(s') \cap |A|$, by Lemma 22 (one way). But condition TYFT *bisimilarity* of Definition 19 gives us a TYFT proof also for $t \xrightarrow{l} t'$, with $t' \equiv_E s'$, using the same premises. Since $\overline{v^Z}(t') \cap |A| = \overline{v^Z}(s') \cap |A|$, it follows that $\langle l, a\rangle$ is also in $\bar{v}_{P_A}(t)$ by Lemma 22 (the other way). Similarly, every element in $\bar{v}_{P_A}(t)$ is in $\bar{v}_{P_A}(s)$. □

Therefore under the mentioned condition on the equations P^Δ maps Γ-algebras to Γ-algebras. To prove that it is a well-defined functor, we have to show that if $g : A \to B$ is a Γ-homomorphism, then so is $P^\Delta(g)$. We do this in the next proposition, by exploiting the fact that only TYFT rules in the theory of Δ are considered in the definition of P^Δ.

Proposition 24 (P^Δ is a well-defined functor). *Let Δ be as SOS specification such that the equations satisfy the conditions of Proposition* 23. *Then P^Δ is a well-defined endo-functor on $\mathscr{A}lg(\Gamma)$.*

Proof. We have to prove that if $g : A \to B$ is a Γ-homomorphism, then so is $P^{\Delta}(g) : P^{\Delta}(A) \to P^{\Delta}(B)$. Recall that by Definition 19 $P^{\Delta}(g)$ is defined, for all $T \subseteq |P_A|$, as $\mathscr{P}^{\Delta}(g)(T) = \{g(a) \mid a \in T \cap |A|\} \cup \{\langle l, g(a)\rangle \mid \langle l, a\rangle \in T \cap L \times |A|\}$.[9] Let us denote $P^{\Delta}(g)$ by $\hat{g}$ for the rest of the proof.

The proof exploits the characterization of $P^{\Delta}(A)$ (and $P^{\Delta}(B)$) as limits of corresponding chains showing by induction that for all $k \in \mathbb{N}$, $\hat{g} : P_k^A \to P_k^B$ is a well-defined homomorphism. For $k = 0$, this is obvious. For the inductive case, let us assume that $\hat{g} : P_k^A \to P_k^B$ is a Γ-homomorphism. We have to show that for every $f \in \Sigma_n$ and for every $T_1, \dots, T_n \subseteq (L \times |A| + |A|)$,

$$\hat{g}(f^{P_{k+1}^A}(T_1, \dots, T_n)) = f^{P_{k+1}^B}(\hat{g}(T_1), \dots, \hat{g}(T_n)). \tag{2}$$

Observe that the two sets in this equations are subsets of $L \times |B| + |B|$: the fact that their intersections with $|B|$ are equal easily follows from Fact 16. Therefore we can focus on the elements of $L \times |B|$ that they contain.

Let us consider the left-to-right inclusion of (2) first. Suppose that $\langle l, b\rangle \in \hat{g}(f^{P_{k+1}^A}(T_1, \dots, T_n))$, or, equivalently, that there exists an $a \in A$ such that $g(a) = b$ and $\langle l, a\rangle \in f^{P_{k+1}^A}(T_1, \dots, T_n)$. Then, by the definition of $f^{P_{k+1}^A}$ in Proposition 20 either $\langle l, a\rangle \in f^{P_k^A}(T_1, \dots, T_n)$, and in this case $\langle l, b\rangle \in f^{P_k^B}(\hat{g}(T_1), \dots, \hat{g}(T_n)) \subseteq f^{P_{k+1}^B}(\hat{g}(T_1), \dots, \hat{g}(T_n))$ by induction hypothesis. Or there exists a TYFT rule $r \in Th(\Delta)$ and a Y-linear assignment v such that $\langle P_k^A, v\rangle$ is a solution to the premises and satisfies a few other conditions. Now let $w : \mathscr{V}(r) \to |P_B|$ be the assignment defined as $w = \hat{g} \circ v$. It is not difficult to check that $\langle P_k^B, w\rangle$ is a solution to the premises of r, and that it satisfies the conditions ensuring that $\langle l, g(a)\rangle \in f^{P_{k+1}^B}(\hat{g}(T_1), \dots, \hat{g}(T_n))$, which proves the left-to-right inclusion of (2).

For the right-to-left inclusion of (2), suppose that $\langle l, b\rangle \in f^{P_{k+1}^B}(\hat{g}(T_1), \dots, \hat{g}(T_n))$. Then, by the *Inductive Case* in Proposition 20, either $\langle l, b\rangle \in f^{P_k^B}(\hat{g}(T_1), \dots, \hat{g}(T_n))$ and we can conclude easily. Or there exists a TYFT rule $r \in Th(\Delta)$, $r = \frac{\{t_i \xrightarrow{l_i} y_i\}_{i \in I}}{f(x_1, \dots, x_n) \xrightarrow{l} t}$, there exists a Y-linear assignment $w : \mathscr{V}(r) \to |P_B|$ such that:

(a) $\forall i \in I . \langle P_k^B, w\rangle \models \{t_i \xrightarrow{l_i} y_i\}_{i \in I} \wedge$
(b) $\forall j \in \{1, \dots, n\} . w(x_j) = \hat{g}(T_j) \wedge$
(c) $b \in \bar{w}_{P_k^B}(t) \cap |B|$.

We have to show, in this last case, that $\langle l, b\rangle \in \hat{g}(f^{P_{k+1}^A}(T_1, \dots, T_n))$. The main point here is to determine a Y-linear assignment $v : \mathscr{V}(r) \to |P_A|$ such that all the premises of the rule r are satisfied in P_k^A. Firstly, let (a′) $v(x_j) = T_j$ for all $j \in \{1, \dots, n\}$. Since all x_j are distinct variables, this is well-defined, and we have $\hat{g}(v(x_j)) = \hat{g}(T_j) = w(x_j)$. Next we define v on the variables in $\{y_i\}_{i \in I}$. From Definition 10 we know that for all $i \in I$, $\mathscr{V}(t_i) \subseteq \{x_1, \dots, x_n\}$, therefore $\bar{v}_{P_k^A}(t_i)$ is well defined, and we have $\hat{g}(\bar{v}_{P_k^A}(t_i)) = \bar{w}_{P_k^B}(t_i)$, because $\hat{g} : P_k^A \to P_k^B$ is a homomorphism by hypothesis. Now from (a) we know

[9] Notice that $\mathscr{P}^{\Delta}(g)$ is uniquely determined in this way by the fact that $\mathscr{P}^{\Delta}$ is required to be a lifting of P_L along $|\_|$.

that for all $i \in I$, $\exists b_i. \in w(y_i) \cap |B| \wedge \langle l_i, b_i \rangle \in \bar{w}_{P_k^B}(t_i) = \hat{g}(\bar{v}_{P_k^A}(t_i))$. This implies that there is an $a_i \in |A|$ such that $g(a_i) = b_i$ and $\langle l_i, a_i \rangle \in \bar{v}_{P_k^A}(t_i)$. Define $v(y_i) = \{a_i\}$: we clearly have (b') $\langle P_k^A, v \rangle \models t_i \xrightarrow{l_i} y_i$, and also $w(y_i) \cap |B| = \hat{g}(v(y_i) \cap |A|)$. Finally, from (c) above $(b \in \bar{w}_{P_k^B}(t) \cap |B|)$ we get $b \in \hat{g}(\bar{v}_{P_k^A}(t)) \cap |B|$, which implies that there is an $a \in |A|$ such that $g(a) = b$ and $a \in \bar{v}_{P_k^A}(t)$. From these last observations together with facts (a') and (b') we deduce that $\langle l, a \rangle \in f^{P_{k+1}^A}(T_1, \dots, T_n)$, and thus $\langle l, b \rangle = \langle l, g(a) \rangle \in \hat{g}(f^{P_{k+1}^A}(T_1, \dots, T_n))$, which concludes the proof. □

Example 25 (*Endofunctor lifting*). The lifting derived from the rules of structured transition systems over commutative monoids (cf. Proposition 7) coincides, but for minor details, with the power monoid construction presented in [5], i.e.,

$$\varepsilon^{P^\Delta(A)} = \{\varepsilon^A\}$$

$$S_1 \otimes^{P^\Delta(A)} S_2 = \{\langle l_1 \otimes^L l_2, b_1 \otimes^A b_2 \rangle \mid \langle l_i, b_i \rangle \in S_i\} \cup \{b_1 \otimes^A b_2 \mid b_i \in S_i\}.$$

Concerning the π-calculus example of Example 8, the ACI equations for $|$ and $+$ are easily shown to satisfy the TYFT bisimilarity condition of Definition 19, but the third substitution axiom in the specification (call it [sub3]) provides a counterexample to that condition: The rule [out] allows to derive a sequent

$$\bar{z} y.P[z/x] \xrightarrow{\bar{z} y} P[z/x] \tag{3}$$

whose source is equivalent to $(\bar{x} y.P)[z/x]$ but there is no sequent $(\bar{x} y.P)[z/x] \xrightarrow{\bar{z} y} Q$ with $Q \equiv_E P[z/x]$ because there exists no rule for substitution.

3.3. Structured coalgebras and compositionality

Next we shall analyse the constraints imposed by the structured coalgebra setting on the behaviour of a system. We shall see that this representation only works for systems which are compositional in the sense of Example 9. In fact, the next proposition shows that what we have called "compositionality of behaviour" is precisely captured by the homomorphism property of the coalgebra structure map.

Proposition 26 (Homomorphism property of coalgebra structure). *Let $\Delta = \langle \Gamma, L, X, R \rangle$ be a SOS specification with $\Gamma = \langle \Sigma, E \rangle$ satisfying the condition of Definition* 19, *and let $P^\Delta : \mathscr{A}lg(\Gamma) \to \mathscr{A}lg(\Gamma)$ be the corresponding lifting of the endofunctor P_L. Let hts be a heterogeneous transition system over Γ, L, and let $\sigma_{hts} : A \to P_\Delta(A)$ be defined as $\sigma_{hts}(a) = \{\langle l, b \rangle \mid a \xrightarrow{l}_{hts} b\} \cup \{a\}$.*

Then σ_{hts} is a Γ-homomorphism if and only if for every transition $f^A(a_1, \dots, a_n) \xrightarrow{l}_{hts} b$ in the system there is a TYFT *proof of a sequent $f(x_1, \dots, x_n) \xrightarrow{l} t$ with premises $\Phi = \{x_j \xrightarrow{m} z_{x_j,m,c} | j \in \{1, \dots, n\} \wedge a_j \xrightarrow{m}_{hts} c\}$ and such that $b \in \bar{w}(t)$, where the assignment $w : \mathscr{V}(\Phi) \to |P_A|$ is defined as $w(x_j) = \sigma_{hts}(a_j)$ for all $j \in \{1, \dots, n\}$, and $w(z_{x,m,c}) = c$.*

Proof. Note that we have:

$$
\begin{array}{ll}
f^A(a_1,\dots,a_n) \xrightarrow{l}_{hts} b & \text{iff [by definition of } \sigma_{hts}],\\
\langle l,b\rangle \in \sigma_{hts}(f^A(a_1,\dots,a_n)) & \text{iff [by homomorphism property of } \sigma_{hts}],\\
\langle l,b\rangle \in f^{P^\Delta(A)}(\sigma_{hts}(a_1),\dots,\sigma_{hts}(a_n)) & \text{iff [introducing an assignment } v],\\
\langle l,b\rangle \in \bar{v}(f(x_1,\dots,x_n)) \wedge \forall j \in \{1,\dots,n\}\,.\,v(x_j) = \sigma_{hts}(a_j). &
\end{array}
$$

Now the statement follows immediately by applying Lemma 22 to the last formula. □

In other terms, a transition $f^A(a_1,\dots,a_n) \xrightarrow{l}_{hts} b$ out of a composed state exists if and only if there is a TYFT proof for a corresponding sequent using some transitions out of $a_1,\dots a_n$ to satisfy the premises. That means, only such transition systems where all transitions can be derived using TYFT proofs from the specification can be represented as coalgebras. In the theory of algebraic specification such a condition corresponds to the notion of (term-) generated algebra, i.e., an algebra where the initial homomorphism is surjective. In Horn clause logic (or Logic Programming), this is nothing else but the well-known closed world assumption. As a consequence we have the necessary condition that the SOS specification Δ is equivalent to the set of all TYFT rules in the theory $Th(\Delta)$, that is, more complex rules have to be derivable from more basic ones.

However, due to the presence of equations, this condition is not sufficient. In fact, Example 9 can be used to show that the condition of Proposition 26 is not satisfied in the π-calculus fragment we are considering: Although the specification consists entirely of rules in deSimone format, it is not compositional in the sense of Proposition 26 since there exists a τ-transition from $P[u/v]$, but there is no derived TYFT rule with a consequence of the form $s[u/v] \xrightarrow{l} t$ by which the transition from $P[u/v]$ could be proved. It has been stressed that the (non-)compositionality of the behaviour is closely related to the fact that bisimulation is (not) a congruence. In fact, it can be shown that whenever there is a lifting P^Δ of functor P_L to $\mathscr{A}lg(\Gamma)$ such that a system is representable as a P^Δ-coalgebra, then its coarsest bisimulation is a congruence. Therefore, in a heterogeneous transition system over Γ, the observation that bisimilarity is not a congruence is already sufficient to show that it cannot be represented as a coalgebra for *any* lifting of the functor P_L to Γ-algebras.

Proposition 27 (Bisimilarity is a congruence in structured coalgebras). *Let Γ be an algebraic specification, L be a Γ-algebra of labels, and $B_L^\Gamma : \mathscr{A}lg(\Gamma) \to \mathscr{A}lg(\Gamma)$ be a lifting of $P_L : \mathbf{Set} \to \mathbf{Set}$. If $\langle S, \sigma\rangle$ is a B_L^Γ-coalgebra and $\langle S, \to\rangle$ its corresponding structured LTS, then bisimilarity on $\langle S, \to\rangle$ is a congruence with respect to the operators in Γ.*

Proof. By Proposition 28 below, the right adjoint of Proposition 13 lifts to a right adjoint $R^\Gamma : \mathscr{A}lg(\Gamma) \to B_L^\Gamma$-**Coalg**. Thus, B_L^Γ-**Coalg** inherits a final object $R^\Gamma(\mathbf{1})$ from $\mathscr{A}lg(\Gamma)$. Since V_B^Γ is a right adjoint as well, the final object is preserved. Hence, bisimilarity induced by the final morphism to $R^\Gamma(\mathbf{1})$ in B^Γ-**Coalg** is determined by the underlying sets and functions, i.e., its definition does not use the algebraic structure of states and transitions. Since the final morphisms in B_L^Γ-**Coalg** are Γ-homomorphisms, it follows that bisimilarity is a congruence. $\square$

Proposition 28 (Lifting adjunctions). *Let Γ be a specification, $B : \mathbf{Set} \to \mathbf{Set}$ be a functor, and $B^\Gamma : \mathscr{A}lg(\Gamma) \to \mathscr{A}lg(\Gamma)$ be a lifting of B along V^Γ. Then, the forgetful functor $V_B^\Gamma : B^\Gamma$-**Coalg** $\to B$-**Coalg** defined on objects and arrows by*

$$\langle a : A \to B^\Gamma A \rangle \mapsto \langle V^\Gamma a : V^\Gamma A \to V^\Gamma B^\Gamma A = B V^\Gamma A \rangle \quad \textit{and} \quad f \mapsto V^\Gamma f$$

*has a left adjoint $F_B^\Gamma : B$-**Coalg** $\to B^\Gamma$-**Coalg** with $U^\Gamma \circ F_B^\Gamma = F^\Gamma \circ U$, denoting by U : B-**Coalg** $\to$ **Set** and $U^\Gamma : B^\Gamma$-**Coalg** $\to \mathscr{A}lg(\Gamma)$ the obvious underlying functors.*

*Moreover, if $U : B$-**Coalg** $\to$ **Set** has a right adjoint $R : \mathbf{Set} \to B$-**Coalg** this lifts to a right adjoint $R^\Gamma : \mathscr{A}lg(\Gamma) \to B^\Gamma$-**Coalg** for U^Γ with $R \circ V^\Gamma = V_B^\Gamma \circ R^\Gamma$.*

$$\begin{tikzcd}
B\text{-}\mathbf{Coalg} \arrow[r, bend left, "F_B^\Gamma"] \arrow[d, bend left, "U"] & B^\Gamma\text{-}\mathbf{Coalg} \arrow[l, bend left, "V_B^\Gamma"] \arrow[d, bend left, "U^\Gamma"] \\
\mathbf{Set} \arrow[u, bend left, "R"] \arrow[r, bend left, "F^\Gamma"] & \mathcal{A}lg(\Gamma) \arrow[l, bend left, "V^\Gamma"] \arrow[u, bend left, "R^\Gamma"]
\end{tikzcd}$$

Proposition 28 is proved in [6] using techniques analogous to those used in [29]. In fact, it can be shown that the category B^Γ-**Coalg** of coalgebras over Γ-algebras is isomorphic to a category of *bialgebras* in the sense of [29].

4. Compositional interpretation of open systems

The discussion in the previous section shows that a transition system where bisimilarity $\sim$ is not a congruence cannot be represented as a structured coalgebra. Hence, the idea is to modify the system in such a way that bisimilarity in the new system coincides with the coarsest bisimulation which is a congruence in the original system. In [26] such an equivalence has been characterised operationally as *dynamic bisimilarity*.

The basic idea of dynamic bisimulation is to allow at every step of bisimulation not only the execution of an action, but also the embedding of the two agents under measurement within the same, but otherwise arbitrary, context. As stressed in Section 1, this notion of bisimulation is very natural for open systems, which have to be compared also with respect to their behaviour in response to dynamic reconfigurations like the addition of new components.

The following definition is made parametric with respect to the set of allowed contexts.

Definition 29 (*Dynamic bisimulation*). Let $hts = \langle S, \rightarrow \rangle$ be a heterogeneous transition system over Γ and L, $\mathscr{C}$ be a set of contexts over Γ, and let $\mathscr{R}$ be a binary relation over S.

Then $\Phi_d^{\mathscr{C}}$, a function from relations to relations, is defined as follows:

$(s,t) \in \Phi_d^{\mathscr{C}}(\mathscr{R})$ if and only if for each $l \in L$ and for each context $C \in \mathscr{C}$:

- whenever $C[s] \xrightarrow{l} s'$ there exists t' such that $C[t] \xrightarrow{l} t'$ and $(s',t') \in \mathscr{R}$;
- whenever $C[t] \xrightarrow{l} t'$ there exists s' such that $C[s] \xrightarrow{l} s'$ and $(s',t') \in \mathscr{R}$.

A relation $\mathscr{R}$ is called *$\mathscr{C}$-dynamic bisimulation* if and only if $\mathscr{R} \subseteq \Phi_d^{\mathscr{C}}(\mathscr{R})$. It is called *dynamic bisimulation* if $\mathscr{C}$ is the set of all contexts over Γ.

The relation $\sim_{\mathscr{C}}^d = \cup \{\mathscr{R} \mid \mathscr{R} \subseteq \Phi_d^{\mathscr{C}}(\mathscr{R})\}$ is called *$\mathscr{C}$-dynamic bisimilarity*. It is called *dynamic bisimilarity* $\sim^d$ if $\mathscr{C}$ is the set of all contexts over Γ.

A set of contexts $\mathscr{U}$ over Γ is called *universal for hts* if $\sim_{\mathscr{U}}^d = \sim^d$.

It is shown in [26] that dynamic bisimilarity is the coarsest bisimulation which is a congruence. Therefore it coincides with observational congruence if (and only if) $\approx$ is a bisimulation. For example, $\sim^d$ and $\approx$ are different for CCS with weak bisimulation [23], which is the main case study in [26], because for this process algebra $\approx$ is not a bisimulation; instead they coincide for structured transition systems as well as for the π-calculus fragment we are considering.

It turns out that the idea of dynamic bisimulation – to consider a system as acting in an open, dynamic environment – shall also solve our problem of presenting the running example as structured coalgebras. For this purpose we will extend the transition relation by what one could call “conditional transitions”, i.e., transitions that would be enabled if a certain context would be provided. Instead of defining the modification directly on the transitions, in the definition below we add appropriate introduction and elimination rules to the SOS specification.

Definition 30 (*Closure under context transitions*). Given a SOS specification $\Delta = \langle \Gamma, L, X, R \rangle$ where all rules in R are in separated format (see Definition 10), let T_Δ be the corresponding initial transition system, and $\mathscr{U}$ be a universal set of contexts for T_Δ satisfying the following properties.

(1) $\mathscr{U}$ is generated by a set of basic contexts $C_i = op(x_1, \ldots, x_{i-1}, \bullet, x_{i+1}, \ldots, x_n)$ for $op \in \Sigma_n$

(2) for every equation $s = s'$ in Δ and every substitution $\sigma : X \rightarrow T_\Sigma(X)$ such that there exists a TYFT proof of sequent $\bar{\sigma}(s) \xrightarrow{l} t$ with premises $\{t_i \xrightarrow{l_i} y_i\}_{i \in I}$ and no TYFT proof of sequent $\bar{\sigma}(s') \xrightarrow{l} t$ with the same premises, there exists $D \in \mathscr{U}$ and $f \in \Sigma_n$ such that $s' = D[f(x_1, \ldots, x_n)]$.

Then the *closure of Δ under context transitions* is the specification

$$\Delta^* = \langle \Gamma, \mathscr{U} \times L, X, R^* \rangle$$

which is derived as follows:

Relabelling: The pairs $\langle C, l \rangle$ of contexts and labels shall be denoted as $C \vdash l$.

Replace each rule

$$\frac{\{s_i \xrightarrow{l_i} t_i\}_{i\in I}}{s \xrightarrow{l} t} \quad \text{by} \quad \frac{\{s_i \xrightarrow{\bullet\vdash l_i} t_i\}_{i\in I}}{s \xrightarrow{\bullet\vdash l} t}$$

by adding the empty context $\bullet$. This empty context shall usually be omitted.

Context introduction: For all $s, s' \in T_\Sigma(X)$ with $\mathscr{V}(s') \subseteq \mathscr{V}(s)$, and for all $C, D \in \mathscr{U}$ such that $D[s'] \equiv_E s$ introduce a rule

$$[+D] \quad \frac{s \xrightarrow{C\vdash l} y}{s' \xrightarrow{C\circ D\vdash l} y}$$

with $y \notin \mathscr{V}(s)$.

Context elimination: For all $C, D \in \mathscr{U}$ introduce a rule

$$[-D] \quad \frac{x \xrightarrow{C\circ D\vdash l} y}{D[x] \xrightarrow{C\vdash l} y}$$

The two new families of rules above represent two kinds of operations on transitions. The context introduction rules allow a process to "borrow" a context C in order to perform a transition labelled by l. This debt is recorded in the label of the new transition as $C \vdash l$. With the context elimination rules, the context is given back, deriving in this way the original transition. Conditions (1) and (2) on contexts ensure that the closure adds to the system "enough" tyft rules so that all transitions can be derived using tyft rules only.

Example 31 (*Closure under context*). Next we shall demonstrate how, in the case of our π-calculus example, the context completion allows us to satisfy both the tyft bisimilarity condition in Definition 19 and the condition of Proposition 26 ensuring the homomorphism property of the coalgebra map.

It is well-known (see e.g. [24]) that substitutions provide a set of universal contexts for the π-calculus, i.e., $\mathscr{U} = \{\bullet[x/y] \mid x, y \in \mathscr{N}\}$. This obviously satisfies the second condition in Definition 30 because in all axioms which do not already satisfy the tyft bisimilarity condition (i.e., the substitution axioms) the left-hand side is of the form $f(x_1, \ldots, x_n)[x/y]$.

In Example 25 we have shown that, e.g., the third substitution axiom [sub3] in the specification of Example 8 provides a counterexample to the tyft bisimilarity condition using transition $\bar{z}y.P[z/x] \xrightarrow{\bar{z}y} P[z/x]$ (3). After completing the specification according to Definition 30 we can derive a tyft rule

$$[r\text{-}sub3] \quad \frac{\bar{z}y.P[z/x] \xrightarrow{\bar{z}y} Q}{\bar{x}y.P \xrightarrow{\bullet[z/x]\vdash\bar{z}y} Q}$$

using axiom [sub3] and the context introduction rule $[+\bullet[z/x]]$. Now, using [r-sub3] and the context elimination rule $[-\bullet[z/x]]$, from (3) we can derive the transition $(\bar{x}y.P)[z/x] \stackrel{\bar{z}y}{\to} Q$ required to satisfy the TYFT bisimilarity condition of Proposition 23.

The deduction rules of Definition 30 may also be applied in order to decompose rules with non-basic source, which violate the condition of Proposition 26. From the π-calculus rules [*in*], [*out*], and [*com*] we can derive the rule

$$[tau] \quad \frac{}{\bar{u}y.P|u(z).Q \stackrel{\tau}{\to} P|(Q[y/z])}$$

Since $\bar{u}y.P|u(z).Q = (\bar{u}y.P|v(z).Q)[u/v]$ (assuming that $v \notin \text{free-names}(P,Q)$) with $C[\bullet] = \bullet[u/v]$ and $s' = \bar{u}y.P|v(z).Q$ we obtain the context introduction and elimination rules

$$[+\bullet[u/v]] \quad \frac{\bar{u}y.P|u(z).Q \stackrel{\tau}{\to} R}{\bar{u}y.P|v(z).Q \stackrel{\bullet[u/v]\vdash\tau}{\to} R}, \quad [-\bullet[u/v]] \quad \frac{P \stackrel{\bullet[u/v]\vdash\tau}{\to} Q}{P[u/v] \stackrel{\tau}{\to} Q}.$$

With these rules it is possible to derive the τ-transition of Example 9 from a transition out of the agent P as follows.

$$\cfrac{\cfrac{\cfrac{}{\bar{u}y.0|u(z).0 \stackrel{\tau}{\to} 0|0 = 0}\ [tau]}{\bar{u}y.0|v(z).0 \stackrel{\bullet[u/v];\tau}{\to} 0|0 = 0}\ [+\bullet[u/v]]}{\bar{u}y.0|u(z).0 \stackrel{\tau}{\to} 0}\ [-\bullet[u/v]]$$

The following proposition gives the semantical justification of the closure construction and states that the original problem, the coalgebraic presentation of the heterogeneous transition system generated by the rules, is solved for systems satisfying the conditions of Definition 29. The proof is based on the observation that, after context completion, every derivation of a transition can be transformed into a derivation using only the TYFT rules in the theory of Δ.

Proposition 32 (Dynamic bisimulation). *Let $\Delta = \langle \Gamma, L, X, R\rangle$ be an SOS specification with rules in separated format, such that its closure under context transitions Δ^* is defined. Then, dynamic bisimilarity on the initial Δ-transition system T_Δ coincides with bisimilarity on the initial Δ^*-transition system $T_{\Delta^*} = \langle T_{\Gamma^*}, \to_{\Delta^*}\rangle$.*

*Moreover, $P_L : \mathbf{Set} \to \mathbf{Set}$ lifts to an endofunctor $P^{\Delta^*} : \mathscr{A}lg(\Gamma) \to \mathscr{A}lg(\Gamma)$, and the P_L-coalgebra $\langle |T_{\Gamma^*}|, \tau_{\Delta^*}\rangle$ corresponding to the transition system T_{Δ^*} lifts to a P^{Δ^*}-coalgebra $\langle T_{\Gamma^*}, \tau_{\Delta^*}\rangle$ which is initial in P^{Δ^*}-**Coalg**.*

Proof. We show that, given a set of transitions, each proof of a transition using the rules in Δ^* can be transformed into a proof using only TYFT rules in the theory of Δ^*. This implies that Δ^* satisfies the TYFT bisimilarity condition of Definition 19 and also that T_{Δ^*} satisfies the condition of Proposition 26.

In general, when deriving a new transition from a couple of given ones, we may either use a rule or an equation. In both cases we have to show that the same effect can be obtained using only TYFT rules and no equations in the theory. In the first case, suppose that the rule applied is

$$[r] \quad \frac{\{t_i \xrightarrow{l_i} y_i\}_{i\in I}}{op(s_1,\dots,s_n) \xrightarrow{l} t}$$

which by hypothesis is in separated format (but not necessarily TYFT). The closure under context transitions yields rules

$$[+C_i] \quad \frac{op(s_1,\dots,s_n) \xrightarrow{l} y}{s_i \xrightarrow{C_i \vdash l} y} \quad \text{and} \quad [-C_i] \quad \frac{x \xrightarrow{C_i \vdash l} y}{C_i[x] \xrightarrow{l} y}$$

for all $i \in \{1,\dots,n\}$ and $C_i = op(s_1,\dots,s_{i-1},\bullet,s_{i+1},\dots,s_n)$. Now, composing rule $[r]$ with $[+C_i]$ we obtain

$$[r; C_i] \quad \frac{\{t_i \xrightarrow{l_i} y_i\}_{i\in I}}{s_i \xrightarrow{C_i \vdash l} t}$$

which, followed by $[-C_i]$, allows all the deductions possible using $[r]$. Notice that the number of symbols in both s_i and $C_i[x]$ is strictly smaller than in $op(s_1,\dots,s_n)$. Thus, successive decomposition of proof steps finally yields a proof using TYFT rules only.

In the second case assume that a transition is derived by using an axiom $[ax]$ $s=t$. The application of $[ax]$ to a state, substituting an instance of s by an instance of t, can be mimicked by the rules

$$[r\text{-}ax] \quad \frac{s \xrightarrow{l} y}{t \xrightarrow{l} y} \quad (\text{for all } l \in L)$$

which are themselves derived by applying $[ax]$ to the tautological rule having $s \xrightarrow{l} y$ both as premise and conclusion. Now, using the same transformations as above, every proof step using $[r\text{-}ax]$ can be decomposed into a sequence of steps using TYFT rules only. Moreover, by initiality of T_{Γ^*} in $\mathscr{A}lg(\Gamma^*)$, the coalgebra structure τ_{Δ^*} is the unique homomorphism into $P^{\Delta^*}(T_{\Gamma^*})$ which is therefore the only coalgebra structure on T_{Γ^*}. Now, it is easy to show using the techniques of [29] that P^{Δ^*}-**Coalg** has an initial coalgebra whose carrier is the initial algebra T_{Γ^*}. This implies that T_{Δ^*} is this initial coalgebra.

It remains to show that dynamic bisimilarity $\sim^d$ on T_Δ coincides with bisimilarity $\sim$ on T_{Δ^*}. Consider states $s,t \in T_\Gamma$. It is easy to see that T_Γ and T_{Γ^*} have isomorphic carriers. Thus, without loss of generality, we assume that they are equal, i.e., $s,t \in T_{\Gamma^*}$. First we show that $s \sim^d t$ implies $s \sim t$. Assume $s \xrightarrow{l} s'$. By $s \sim^d t$ we have that $C[s] \xrightarrow{l} s'$ implies $C[t] \xrightarrow{l} t'$ for each context C. Using the empty context $C = \bullet$ this yields $t \xrightarrow{l} t'$ as required. The symmetric argument shows that $s \sim t$. Vice versa, in order to show that $s \sim t$ implies $s \sim^d t$, assume a context C such that $C[s] \xrightarrow{l} s'$. Since $s \sim t$ and

$\sim$ is a congruence (cf. Proposition 27), this implies $C[s] \sim C[t]$ and thus $C[t] \xrightarrow{l} t$. By symmetry we have $s \sim^d t$. $\square$

Example 33 (*Counter example revisited*). Applying the context closure of Definition 30, also the counterexample of Example 9 does not apply anymore. In fact, the two agents $P = \bar{u}y.0|v(z).0$ and $Q = \bar{u}y.v(z).0 + v(z).\bar{u}y.0$ are not bisimilar in the first place. This is due to the additional $\bullet[u/v];\tau$-labelled transition out of P which cannot be matched by a transition of Q.

5. Conclusion

In this paper we have studied the relationship between SOS specifications with structural axioms, transition systems with algebraic structure, and coalgebras in categories of algebras. In particular we have characterised those transition systems for which a structured coalgebraic presentation is possible, and the classes of SOS specifications generating such "well-behaved" systems (cf. Definition 19 and Proposition 26). It turns out that the conditions which guarantee a coalgebraic presentation of a transition system are very similar to the ones which ensure that bisimilarity is a congruence. Essentially they require that the behaviour of the system is compositional, in the sense that all transitions from complex states can be derived using the rules from transitions out of component states. In the case without structural axioms, such condition means that each rule in the specification has a basic operation as the source of its conclusion; indeed this is the common point of many SOS formats (see e.g. [2, 9, 16]). With structural axioms, the situation is more complicated since basic operations can be equivalent to complex terms, and complex states may be decomposed into component states in many different ways (cf. Example 9).

We have also proposed a general procedure (cf. Definition 30) which, when applied to a (not necessarily well-behaved) SOS specification, extends the set of rules in such a way that the resulting specification is well-behaved, that is, its generated transition system can be represented as a coalgebra in a category of algebras (see Proposition 32). The idea is to add transitions which may place a process into a context, simulating in this way the definition of dynamic bisimulation ([26], see also Definition 29). Intuitively, this means to consider processes as open systems which may be reconfigured at runtime.

References

[1] G. Berry, G. Boudol, The chemical abstract machine, Theoret. Comput. Sci. 96 (1) (1992) 217–248.

[2] B. Bloom, S. Istrail, A.R. Meyer, Bisimulation can't be traced, J. ACM 42 (1) (1995) 232–268.

[3] A. Corradini, A. Asperti, A categorical model for logic programs: indexed monoidal categories, in: J.W. de Bakker, W.-P. de Roever, G. Rozenberg (Eds.), Proc. REX Workshop 'Semantics: Foundations and Applications', Beekbergen, The Netherlands, 1992, Lecture Notes in Computer Science, Vol. 666, Springer, Berlin, 1993.

[4] A. Corradini, F. Gadducci, A 2-categorical presentation of term graph rewriting, in: Proc. CTCS'97, Lecture Notes in Computer Science, Vol. 1290, Springer, Berlin, 1997.

[5] A. Corradini, M. Große-Rhode, R. Heckel, Structured transition systems as lax coalgebras, in: B. Jacobs, L. Moss, H. Reichel, J. Rutten (Eds.), Proc. 1st Workshop on Coalgebraic Methods in Computer Science, CMCS'98, Lisbon, Portugal, Electronic Notes in Theoretical Computer Science, Vol. 11, Elsevier Science, Amsterdam, 1998. http://www.elsevier.nl/locate/entcs.
[6] A. Corradini, M. Große-Rhode, R. Heckel, An algebra of graph derivations using finite (co-) limit double theories, in: J.L. Fiadeiro (Ed.), Proc. 13th Workshop on Algebraic Development Techniques, WADT'98, Lecture Notes in Computer Science, Vol. 1589, Springer, Berlin, 1999.
[7] A. Corradini, R. Heckel, U. Montanari, in: From SOS specifications to structured coalgebras: How to make bisimulation a congruence, in Proc. 2nd Workshop on Coalgebraic Methods in Computer Science, CMCS'99, Electronic Notes of Theoretical Computer Science, Vol. 19, Elsevier, Amsterdam, 1999.
[8] A. Corradini, U. Montanari, An algebraic semantics for structured transition systems and its application to logic programs, Theoret. Comput. Sci. 103 (1992) 51–106.
[9] R. De Simone, Higher level synchronizing devices in MEIJE–SCCS, Theoret. Comput. Sci. 37 (1985) 245–267.
[10] H. Ehrig, B. Mahr, in: Fundamentals of algebraic specifications 1: equations and initial semantics, EACTS Monographs on Theoretical Computer Science, Vol. 6, Springer, Berlin, 1985.
[11] G.-L. Ferrari, U. Montanari, P. Quaglia, A π-calculus with explicit substitutions, Theoret. Comput. Sci. 168 (1) (1996) 53–103.
[12] J.A. Goguen, J. Meseguer, Correctness of recursive parallel nondeterministic flow programs, JCSS 27 (1983) 268–290.
[13] F. Gadducci, U. Montanari, The tile model, in: G. Plotkin, C. Stirling, M. Tofte (Eds.), Proof, Language and Interaction: Essays in Honour of Robin Milner, MIT press, Cambridge, MA, 2000, to appear. Paper available from http://www.di.unipi.it/~ugo/festschrift.ps.
[14] J. Goguen, G. Malcolm, A hidden agenda, Tech. Report CS97-538, University of California at San Diego, 1997.
[15] J.W. Gray, The category of sketches as a model for algebraic semantics, Contemp. Math. 92 (1989) 109–135.
[16] J.F. Groote, F. Vandraager, Structured operational semantics and bisimulation as a congruence, Inform. and Comput. 100 (1992) 202–260.
[17] R. Heckel, Open graph transformation systems: a new approach to the compositional modelling of concurrent and reactive systems, Ph.D. Thesis, TU Berlin, 1998.
[18] A. Joyal, M. Nielsen, G. Winskel, Bisimulation from open maps, Proc. LICS'93, Full version as BRICS RS-94-7, Department of Computer Science, University of Aarhus, 1994.
[19] C. Lair, Etude générale de la catégorie des esquisses, Esquisses Math. 24 (1974).
[20] J. Meseguer, Membership algebra as logical framework for equational specification, in: F. Parisi Presicce (Ed.), Recent Trends in Algebraic Development Techniques, Lecture Notes in Computer Science, Vol. 1376, Springer, Berlin, 1998, pp. 18–61.
[21] J. Meseguer, U. Montanari, Petri nets are monoids, Inform. and Comput. 88 (2) (1990) 105–155.
[22] M. Meseguer, U. Montanari, Mapping tile logic into rewriting logic, in: F. Parisi-Presicce (Ed.), Recent Trends in Algebraic Development Techniques, Lecture Notes in Computer Science, Vol. 1376, Springer, Berlin, 1998, pp. 62–91.
[23] R. Milner, in: A Calculus for Communicating Systems, Lecture Notes in Computer Science, Vol. 92, Springer, Berlin, 1980.
[24] R. Milner, J. Parrow, D. Walker, A calculus of mobile processes, Inform. and Comput. 100 (1992) 1–77.
[25] U. Montanari, M. Pistore, in: An introduction to history dependent automata, Proc. 2nd Workshop on Higher-Order Operational Techniques in Semantics (HOOTS II), Electronic Notes of Theoretical Computer Science, Vol. 10, Elsevier, Amsterdam, 1998.
[26] U. Montanari, V. Sassone, Dynamic congruence vs. progressing bisimulation for CCS, Fund. Inform. 16 (1992) 171–199.
[27] G. Plotkin, A structural approach to operational semantics, Tech. Report DAIMI FN-19, Aarhus University, Computer Science Department, 1981.
[28] J.J.M.M. Rutten, Universal coalgebra: a theory of systems, Tech. Report CS-R9652, CWI, 1996, Theoret. Comput. Sci., to appear.
[29] D. Turi, G. Plotkin, Towards a mathematical operational semantics, Proc. LICS'97, 1997, pp. 280–305.